170-WP-006-003

Interoperable Catalogues System (ICS) Collection Technical Note (CTN)

Version 1.0

ECS White Paper

July 1996

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

George Percivall /s/ 7/8/96

George Percivall Date
EOSDIS Core System Project

SUBMITTED BY

Steve Fox /s/ 7/8/96

Steve Fox Date
EOSDIS Core System Project

Hughes Information Technology Systems
Upper Marlboro, Maryland

Committee on Earth Observation Satellites
Working Group on Information Systems and Services

Interoperable Catalogues System (ICS) Collection Technical Note (CTN) Version 1.0

CEOS

Working Group on Information Systems and Services

Protocol Task Team

Doc. Ref.: CEOS/WGISS/PTT/CTN

Date: 8 July 1996

Issue: Version 1.0

AUTHORITY

Issue:	Version 1.0
Date:	8 July 1996
Location:	
Author	G. Percivall, see 1.2 for contributors

This document has been approved for publication by the Protocol Task Team of the Committee on Earth Observation Satellites (CEOS) and reflects the consensus of the Protocol Task Team technical panel experts from the CEOS member agencies.

This document is published and maintained by:

Gerhard Triebnig (CEOS PTT Chair)
European Space Agency/ESRIN
Via Galileo Galilei
00044 Frascati
Italy

E-mail: *gerhard.triebnig@esrin.esa.it*

Table of Contents

6. COLLECTION DISCOVERY

7. COLLECTION NAVIGATION

8. COLLECTION SEARCHING

9. COLLECTION NAMES AND LOCATIONS

List of Figures

List of Tables

Document Status Sheet

Version	Date	Comments
0.1	2 April 1996	Initial Draft
0.2	9 May 1996	Draft. Prepared for the May 1996 PTT meeting. Incorporates Comments from CTN Version 0.1.
1.0	9 July 1996	Final for Release B of CIP/ICS. Prepared to support URD, SDD, and CIP Specification developments. Incorporates RID responses from version 0.2 and discussion at May 1996 PTT meeting.

1. Introduction

1.1 Purpose

The need for a Collection Technical Note (CTN) was identified in the February 1996 Protocol Task Team (PTT) meeting in Ottawa. Generally, the Collection TN is to expand on topics in the CIP Specification which relate to collections but are not necessarily related to the protocol. Several specific RIDs were assigned to the CTN during the Ottawa meeting. An appendix of the CTN provides a tracing of the PTT RIDs to sections of the CTN.

After review and approval by the PTT, the contents of the CTN will migrate to several CEOS Interoperable Catalogues System (ICS) documents. The CTN recommends new URD requirements and modification of existing URD requirements. The URD recommendations are specified in an appendix. Some concepts in the CTN will migrate to a new document called the ICS System Design Document (SDD). Material in the CTN on establishing and maintaining collections will end up in a ICS collection maintenance document.

1.2 Development Plan for TN

The CTN is being developed to support Release B of the URD and CIP Specification. Specific milestones for the development of the CTN are listed in Table 1-1.

Table 1-1. Collection TN Milestones

Milestone	Date
Collection TN Version 0.1, Initial Draft	April 1996
Collection TN Version 0.2, Final Draft (Incorporating PTT e-mail comments on Version 0.1)	May 1996 (For discussion at CEOS meeting)
Collection TN Version 1.0, (Incorporating CEOS meeting comments)	July 1996 (To support PTT co-location meeting)
Collection TN Version 1.1 or Migration to other ICS Documents	November 1996

WGISS-PTT members who have contributed to the CTN include the following: George Percivall - NASA/Hughes (lead), Lou Reich - NASA/CSC, the DPRS Team (Stuart Mills ESA/Logica, Steve Smith ESA/Logica, Marco Ferro ESA/ELCA), Simon Marshal ESA/Serco, Graham Bland NASA/EOS, and Nigel Hinds (University of Michigan).

Versions 0.1 and 0.2 of the CTN emphasized background information on the various topics in the CTN and when conclusions were given they are preliminary. These early versions provide background for PTT members in the various areas where conclusions need to be made. Version

1.0 provides conclusions based on PTT discussions. The key step between version 0.2 and version 1.0 was the May PTT meeting.

This document was prepared as part of the ECS Contract between NASA-GSFC-ESDIS and Hughes as described in ECS Engineering Support Directive #17, ECS Extensions Support.

Questions regarding technical information contained within this Paper should be addressed to the following ECS and GSFC contacts:

- ECS Contacts

  George Percivall, Senior Systems Engineer, (301) 925-0368, gperciva@eos.hitc.com

- GSFC Contacts

  Yonsook Enloe, (301) 286-0794, yonsook.enloe@gsfc.nasa.gov

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Hughes Information Technology Systems
1616 McCormick Drive
Upper Marlboro, MD 20774-5372

1.3 Organization of This Document

The chapters of the CTN fall into the following Categories:

- Background and Collection Model

  Chapter 2 provides background on the concept of collections as it exists in the current CIP Specification (Release A), reviews several other data models with similar concepts of collections, and proposes a data model for CIP Collections.

  Chapter 3 provides an estimate on the number of collections of which the ICS might ultimately consist.

- Collection Evolution

  Chapter 4 provides background on the current concepts of collection creation and maintenance. After this refresher, the lifecycle of collections maintained at a Retrieval Manager is described. It is through this description of the lifecycle of collections that maintenance issues are discussed. The Chapter ends with a discussion of maintenance procedures both automated and manual.

- User Scenarios for Collections

  Chapter 5 describes multiple ways in which a user may interact with the ICS Collection Structure. The scenarios are grouped into four methods: Collection Discovery, Collection Navigation, Collection Searching, and Locating Collections with URNs and URLs. The next

four chapters (6, 7, 8, and 9) describe these methods in detail and describes design options for each method.

- Appendices

  There are three appendices. Appendix A provides draft URD requirements based on topics discussed in the CTN. Appendix 2 maps RIDs assigned during the Ottawa meeting to CTN sections. Appendix 3 provides an assignment of CTN ideas to ICS/CIP releases.

1.4 Glossary of Terms in Collection TN

The following terms, as defined below, are used in this document

Directed Graph	A directed graph consists of a set of nodes and a set of arcs. An arc is directed starting at the tail and terminates at the head. (See 2.2.1). Definitions of path and cluster are also provided.
Tree Structure	A tree is a set of nodes, one of which is distinguished as a root, along with a relation ("parenthood") that places a hierarchical structure on the nodes. (See 2.2.2) Definitions of complex trees and uniform trees is provided.
Object Models	(See 2.2.3)
Collection Discovery	With no previous knowledge of existing collections or retrieval managers, a user discovers collections which are of interest.
Collection Navigation	A user has an established context of a given collection and wishes to find related collections (Note collection navigation does not include collection searching)
Collection Searching	A user has a an established context of a given collection and searches the collection tree below that collection for all searches which matches the users query
Collection Locating	The user has a collection name (URN) or a collection location (URL) and by using the name or location, establishes a context to that collection.

1.5 References in Collection TN

The following documents are referenced in this document

305-CD-028	ECS Release B CSMS Segment Communications Subsystem Design Specification
311-CD-002-004	Science Data Processing Segment (SDPS) Database Design and Database Schema Specifications for the ECS Project ECS Document

Number 311-CD-002-004, December 1995. (Note: an updated to this document is currently under development.)

Aho — Data Structures and Algorithms, Alfred Aho, John Hopcroft, and Jeffery Ullman, Addison-Wesley, 1983.

CCSDS — *Time Code Formats*, CCSDS Recommendation for Space Data Systems, Issue 2, CCSDS 301.0-B-2, April 1990, Consultative Committee for Space Data Systems

CEO-ITT — Data and Information Modeling for the CEO Enabling Services, Invitation to Tender, RGC 2/96, CEO Programme, Joint Research Centre, EUROPEAN COMMISSION

CIP-A — Catalogue Interoperability Protocol (CIP) Specification - Release A, Committee on Earth Observation Satellites (CEOS), Doc. Ref. CEOS/WGISS/PTT/CIP-A, 27 March 1996, Issue 1.2

Dao — "Logical Integration of Heterogeneous Databases," Son Dao, Brad Perry and Eddie Shek, Information Sciences Laboratory, Hughes Research Laboratories, Technical Report #597, January 31, 1996.

DCP — "Z39.50 Profile for Access to Digital Collections", Draft six, http://

Dopplick — "Version 1 Data Migration Plan," J. A. Feldman and Thomas Dopplick, EOSDIS Core System (ECS) Project, Technical Paper 160-TP-002-001, January 1995

F&PRS — Functional and Performance Requirements Specification for the Earth Observing System Data and Information Systems (EOSDIS) Core System, Goddard Space Flight Center, National Aeronautics and Space Administration, Document Number 423-41-02.

FGDC — The Federal Geographic Data Committee, Metadata Standards Development, URL: http://fgdc.er.usgs.gov/metahome.html

Gravano — "Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies," Luis Gravano and Hector Garcia-Molina, Computer Science Department, Stanford University, Proceedings of the 21st VLDB Conference, Zurich, Switzerland, 1995. Retrieved from http://gloss.stanford.edu/papers.html.

Hinds — "Managing Metadata for Distributed Information Servers: A Dissertation Proposal," Nigel Hinds, University of Michigan, January 15, 1996.

IETF - comindex — The Common Indexing Protocol Network Working Group, Chris Weider, Bunyip Information Systems, INTERNET-DRAFT, <draft-weider-comindex-00.txt>, March 1995, found at http://services.bunyip.com:8000/products/digger/digger-main.html

IETF-RFC-1737 "Functional Requirements for Uniform Resource Names," K. Sollins (MIT/LCS) and L. Masinter (Xerox Corporation), Network Working Group, Request for Comments: 1737, December 1994

IETF-RFC-1738 Berners-Lee, T., Masinter, L., McCahill, M. (editors), "Uniform Resource Locators (URL)", RFC 1738, December 1994. ftp://ds.internic.net/rfc/rfc1738.txt

IETF-url-irp-04 Uniform Resource Locators for Z39.50 , Internet-Draft draft-ietf-uri-url-irp-04, IETF URI Working Group, Editors: R. Denenberg, J. Kunze, D. Lynch, 5 February 1996

Lycos "Web Agent Related Research at the Center for Machine Translation," Michael L. Mauldin, John R. R. Leavitt, Center for Machine Translation, Carnegie Mellon University, (To be presented at the SIGNIDR meeting, August 4, 1994 in McLean, Virginia) found at http://fuzine.mt.cs.cmu.edu/mlm/signidr94.html

Tomasic, et. al. Data Structures for Efficient Broker Implementation, by Anthony Tomasic, Luis Gravano, Calvin Lue, Peter Schwarz, and Laura Haas (Technical Report, IBM Almaden Research Center, June 1995) Retrieved from http://gloss.stanford.edu/papers.html. Describes GlOSS.

URD-A Catalogue Interoperability Protocol (CIP) - User Requirements Document (URD), CEOS, Doc. Ref.: CEOS/SGISS/PTT/CIP-URD, 27 March 1996, Issue: 1.2

WWWW "GENVL and WWW: Tools for Taming the Web," Oliver A. McBryan, University of Colorado, (To appear in the Proceedings of the First International World Wide Web Conference, ed. O. Nierstasz, CERN, Geneva, May 1994), found at http://www.cs.colorado.edu/home/mcbryan/Home.html.

This page intentionally left blank.

2. Data Model for Collections

This chapter contains a summary of the current definition of collections in the CIP Specification followed by three sections which provide new concepts relating to the CIP collection definition. The three areas of new concepts are meta-collection archetypes, other collection data models, and additional collection types. The last section of the chapter contains suggested enhancements to the current CIP Collection Model based on the new concepts introduced.

2.1 Current CIP Collection Concept

This section provides the current description of collections in the ICS as described in the CIP Specification - Release A[1]. The material is presented here in an abbreviated form to allow ready reference to existing concepts. To find the complete description see CIP Specification.

This section includes the following sub-sections:

- an overview of the collections concept, illustrating the relationships between collections, products and inventories and introducing the notion of hierarchies (Abbreviated version of CIP Specification Section 2.3.1);
- a definition of collection categories, i.e. user theme collection, provider archive collection and provider theme collection (Abbreviated version of CIP Specification Section 2.3.2);
- details of additional collection concepts such as commonality, identifiers and remote members (Abbreviated version of CIP Specification Section 2.3.4).
- CIP Collection Schema (Abbreviated version of CIP Specification Appendix C)

2.1.1 Collection Overview

A collection has members consisting of item descriptors which are product descriptors or other collection descriptors. Additional descriptors such as for guide data are identified for subsequent CIP releases.

As a collection can contain either product descriptors or collection descriptors, it is possible to group a number of collections under a single theme as the data provider or user finds convenient. It is also easy to include an existing set of product descriptors, which are already in an existing collection, in a new collection by just referring to the existing collection name (which will be an attribute of the collection descriptor).

The collection concept is visualized in Figure 2-1 below:

[1] Reference: CIP-A

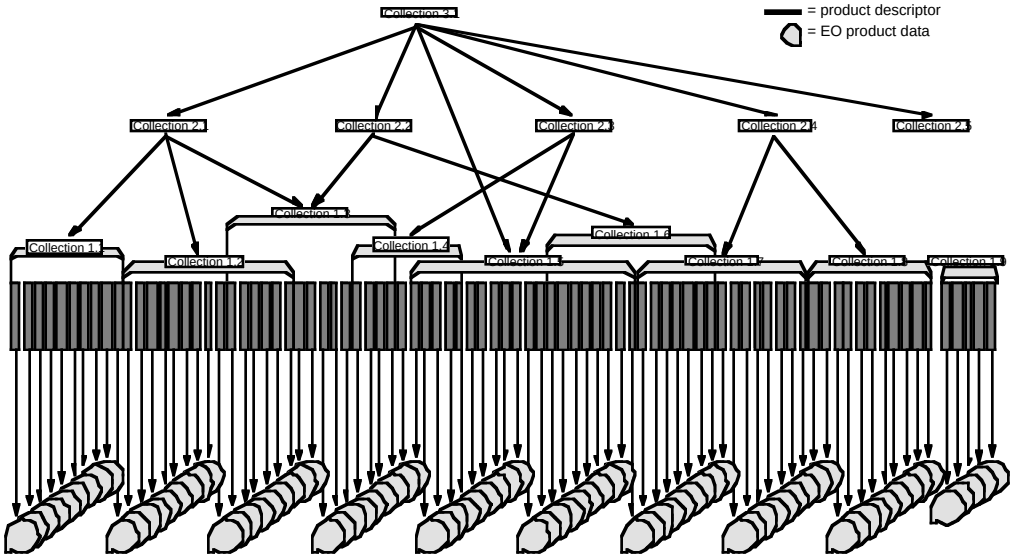

Figure 2-1. The Concept of 'Collection'

The collections in the diagram are numbered so that their relationship can be easily seen, they do not represent the naming of collections in an actual implementation. The terminal collections (labeled '1.x') group the product descriptors (inventory entries) as is appropriate. As can be seen the collections can overlap each other and product descriptors can appear in more than one collection. Above the terminal level collections, there are non-terminal collections that group together any number of other collections. The grouped collections do not all have to be at the same hierarchical level and this grouping of collections can continue to any hierarchical level, with existing collections being included at any other arbitrary level.

2.1.2 Collection Categories

This CIP specification defines three categories or types of collections:

• ***Provider archive collections***: These are collections which are analogous to current physical inventories. The collection includes product descriptors that have uniform attributes. A provider archive collection will be a terminal collection.

• ***Provider theme collections***: these collections are established by data providers grouping together products that have a similar theme, for example, the geographical area covered, the scientific discipline supported by the data, the instrument type, etc. The difference between provider theme and provider archive collections is that archive collections only contain

homogeneous product descriptors, whilst theme collections may have product descriptors of differing formats and attributes. Provider theme collections can be terminal or non terminal dependent on how the data provider decides to organize their data.

• ***User theme collections***: These are collections of potentially quite disparate product descriptors of interest to a relatively small user community researching a particular theme, i.e. in the example, the mid-west flood of 1993. There may be product descriptors from different image archives, in situ measurement archives and bibliographic archives of relevant papers. The members of a thematic collection will in general be formed from the results of a series of filtered searches to build up a set of descriptors. These will then be the target of more focused searches over a period of time, e.g. for more detailed analysis and research. User theme collections can be terminal or non terminal and as stated previously could be envisaged to contain mixed collections (i.e. collection and product descriptors) for future CIP releases.

2.1.3 Collection Concept Details

This section presents a number of collection concepts that need to be taken into account by collection and Retrieval Manager administrators. The definitions are summarized in Table 2-1.

Table 2-1. Summary of Current Collection Concepts

Collection Concept	Summary of Concept
Commonality (consistency)	By definition a collection is a grouping of items that have something in common. Branches lower in the collection tree should have some consistency with its ancestors.
Individuality	A collection member may be a member of two (or more) collections, but duplicate members are not supported within a single collection.
Member Type	A collection member may be a collection descriptor containing a reference to another collection descriptor or another member type, such as product descriptors or guide descriptors.
Collection Trees	As collections can contain pointers to other collections there exists the concept of a 'collection tree' (see Figure 2-1), the leaves of the branches being product or guide descriptors.
Identifier	Each member of a collection (i.e. item descriptor of any type) must have an identifier unique within all the collections in the Retrieval Manager's collection tree.
Uniqueness	By virtue of the unique identifier, every collection existing can be uniquely identified in the domain of all collections (relevant for multiple-site operations).
Remote Members	Normally, a collection tree would be held in one place (say, as a database on a computer). A logical collection tree is where one or more member's collections are held elsewhere - the complete collection tree thus spans multiple sites. Within the complete CIP domain, attributes of collections, products and guide data should only be stored once, and controlled by the Retrieval Manager that owns the items. The only exception to this will be with the members of user theme collections.

2.1.4 CIP Collection Schema

The current metadata to describe a collection is contained in the CIP Abstract Record Structure (Table 2-2) as defined in Appendix C of the CIP Specification.[2]

[2] Reference: CIP-A

Table 2-2. CIP Abstract Record Structure

Collection	=	ItemDescriptorId + (Authoritative) + ItemDescriptorName + 0 { CollectionType + Purpose } 1 + CreationDate + 0 { RevisionDate } n + VersionId + Abstract + (**Review**) + Progress + UpdateFrequency + 0 { AccessConstraints } 1 + 0 { UseConstraints } 1 + **TemporalCoverage** + 1 **SpatialCoverage** }n + DataCentreName + (**DataOriginator**) + (**Investigator**) + (**Technical**) + (ProjectName) + **Keywords** + ArchivingCentreId + 0 { ProcessingCentre } 1 + (ProcessingLevelId + ProcessingLevelDescription) + (StorageMedium) + (DeliveredAlgorithmPackage) + 0 { **CollectionContents** }1 + 0 { LocalityType + LocalityDescription } 1 + (0{**Guide** }n) + 0 { **Browse** }n + (**ReferencePaper**) + 0 { ExternalPublicationCitation } 1 + (**QACollectionStatistics)**

2.2 Collection Structure Models

The previous section describing CIP Collection Concepts is highly dependent on a data structure described as a hierarchical tree. The purpose of this section is to formalize the notion of a tree structure as well as introducing two other models to describe the collection structure. This section describes three modeling approaches: directed graphs, trees, and object modeling. Each section provides a definition and metrics for the model type. Throughout the remainder of the CTN, the model types are used as needed to describe a particular topic, e.g., directed graphs for navigation, trees for searching, object model for collection creation and maintenance.

2.2.1 Directed Graphs

2.2.1.1 Definition of Directed Graphs

A **directed graph** consists of a set of nodes and a set of arcs[3]. An **arc** is directed starting at the **tail** and terminates at the **head**. (See Figure 2-2). We will use nodes to represent collections and arcs to represent relationships between collections, e.g. the head collection is related to the tail collection.. Note that the arcs are uni-directional. The head is related to the tail. And only if there is a second arc in the opposite direction is the tail related to the head

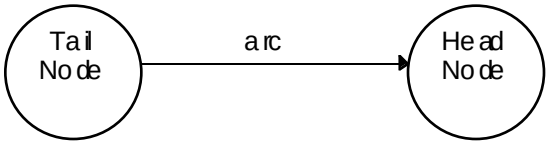

Figure 2-2. Definition of a Directed Graph

A **path** in a directed graph is a sequence of nodes connected by arcs. A path is **simple** if all the nodes on the path, except possibly the first and last are distinct. Note that there is no parent-child relationship in directed graphs and cycles are allowed. A simple **cycle** is a simple path of length at least one that begins and ends at the same vertex.

In a directed graph with multiple nodes, a **cluster** is defined as the set of nodes which are connected by paths. That is, there is a path between any two nodes in a cluster. Also a directed graph may have disjoint clusters which is to say that there is no arcs connecting the clusters.

2.2.1.2 Metrics for Directed Graphs

TBS

2.2.2 Tree Structures

2.2.2.1 Definitions of Tree Structures

A tree is a set of nodes, one of which is distinguished as a root, along with a relation ("parenthood") that places a hierarchical structure on the nodes. Here as with directed graphs, a node will be a collection. Formally, a **tree** can be defined recursively in the following manner[4].

1. A single node by itself is a tree. This node is also the **root** of the tree.

---

[3] Reference: Aho

[4] Reference: Aho

2. Suppose n is a node and T_1, T_2, ..., T_k are trees with roots n_1, n_2, ..., n_k respectively. We can construct a new tree by making n be the **parent** of nodes n_1, n_2, ..., n_k. In this tree n is the root and T_1, T_2, ..., T_k are the **subtrees**of the root. Nodes n_1, n_2, ..., n_k are called the **children** of node n. (See Figure 2-3)

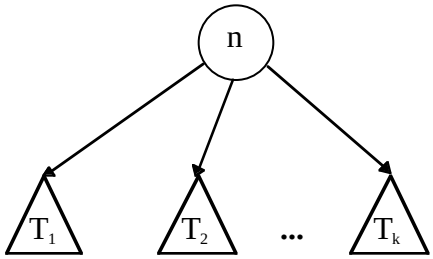

Figure 2-3. Definition of a Simple Tree

It is important note that again the arcs in a tree are directed. That is an arc in a tree defines the parent-child relationship. Furthermore in a simple tree, there are no cycles and therefore no node which its own parent or its own child.

The ICS collection structure will have nodes with more than one parent. The tree in Figure 2-3 has only single parents and is classified as a **simple tree**. A **complex tree** is defined as a tree in which a node may have multiple parents.

For simplicity in estimating the size of the ICS collection structure it is useful to define a **uniform tree** as a simple tree with a uniform number of items per node and each branch of the root node extending to the same depth.

2.2.2.2 Metrics for Trees

Method to calculate the total number of nodes in a uniform tree (provided by Simon Marshall). If a uniform tree is of depth D, with each node having A children, then the total number of nodes T is

$$T = \sum_{i=0}^{D} A^i$$

This expression can be derived by considering the expression for the total number of nodes, T, given A and D for uniform trees with increasing D. The simplest tree has 1 root and A children, i.e.,

For D = 1, T = 1 + A

Adding another layer of descendent nodes adds A nodes for each parent node, i.e.,

For D = 2, T = 1 + A + A*A

Likewise

For D = 3, $T = 1 + A + A^2 + A^3$

By continuing to expand on D, the summation listed above becomes obvious. This equation will be used in the Collection Census Chapter to estimate the number of levels in the collection tree.

2.2.3 Object Models

2.2.3.1 Definition of Object Modeling

It will be useful to discuss collection creation and maintenance to use the relationships defined by object oriented modeling[5]. The inheritance/generalization relationship in particular will be useful when discussing collection structures and the attributes contained in the collections.

Object-oriented methodology organizes a system as a collection of objects, each of which has data structure and behavior and which has meaning within the context of the problem that is being modeled. The following definitions are used in the object models.

- ***Object:*** An abstraction of something in the problem at hand, characterized by a unique name, distinct properties, and well defined behavior.
- ***Class:*** A group of objects with the same meaning, properties ***(attributes)***, behaviors ***(operations)***, and relationships ***(associations)*** with other objects.
- ***Generalization:*** Objects can be generalized into a more generic object class. For example, guides, program descriptions, and general system descriptions could be generalized into a common class called documents. The document class is then called the parent class of guides, program descriptions, and general system descriptions.
- ***Attribute:*** a named property of a class, describing data values held by each object in the class. Classes describe the data property (e.g., color). Each object holds a value (e.g., green) for each attribute defined for the class to which the object belongs.
- ***Operation:*** a part of the behavior of a class. Collectively, all of a class' operations define the things that objects of the class can do.
- ***Link:*** a physical or conceptual connection between object instances -- an instance of an ***association*** (see the next definition).
- ***Association:*** a group of links with common structure and common meaning -- a set of potential links.
- ***Aggregation:*** The model also recognizes a specific kind of relationship, called Aggregation. It indicates that objects of one class (the aggregate) are composed of objects belonging to other classes (the components).

Figure 2-4 shows the notation used by the Object Models. The rectangular boxes in the model denote classes. Each box, shown in full detail, consists of three sections. The name of the class fills the top section, its attributes go in the middle section, and its operations in the bottom

[5] Reference: Rumbaugh

section. Sometimes in high level drawings, only the top section of the box, showing the class name, is shown. A class may be the generalization of several other classes. In Figure 2-4, the "Parent Class" is the generalization of two other classes, each called a "Derived Class." Derived classes always include the attributes and operations provided by their parent classes. The diagrams, therefore, only show any additional attributes or operations which the derived class may have.

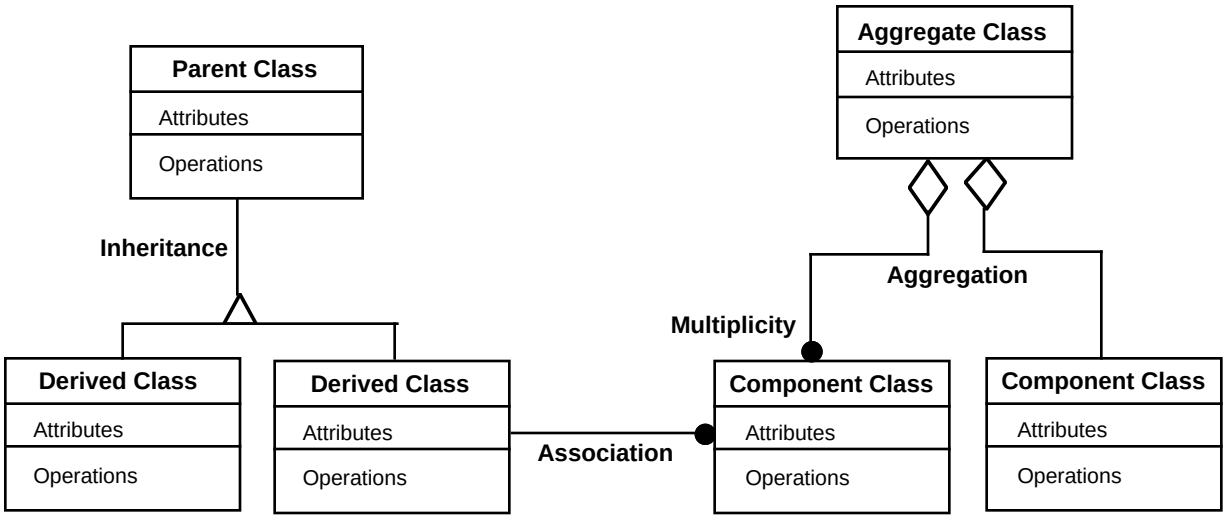

Figure 2-4. Object Model Diagram Notation

Figure 2-4 also shows that there are two classes, each called a "Component Class", have been aggregated into another class, called the "Aggregate Class". There may be design rules which determine how many components of each class an aggregate may have. This is shown by providing an indication of the "Multiplicity" in the diagram. In Figure 2-4, the left component may occur any number of times (zero, one, or many), the right component must occur exactly once. Finally, classes may have relationships, indicated by simple lines. On the design diagrams, they are labeled with the name of the relationship, and they carry an indication of multiplicity.

2.3 Comparison of Collection Models

2.3.1 Current CIP Collection Data Model

Currently the data model for CIP Collection is part of the CIP Domain Object Model[6]. Figure 2-5 shows the collection object and related objects excerpted from the CIP Domain Object model.

[6] Reference: CIP-A, Appendix F

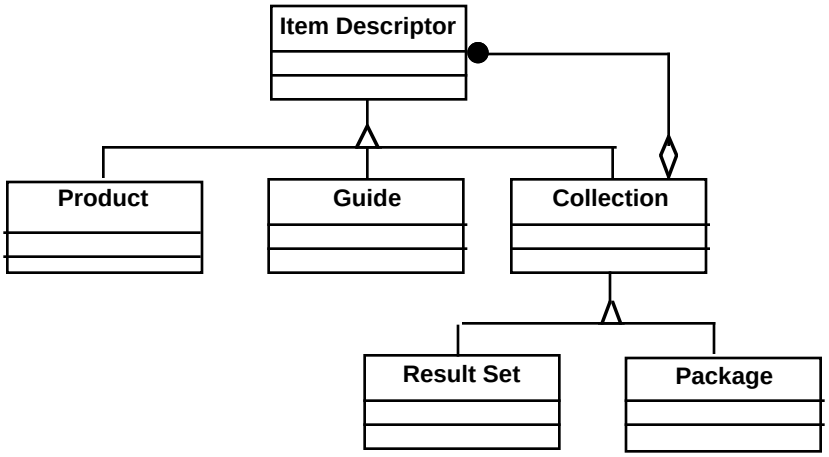

Figure 2-5. Collection Portion of Appendix F: CIP Domain Object Model

2.3.2 Comparison of CIP and Z39.50 Digital Collections Profile

The purpose of this (Section) is to compare the object model implied by the Z39.50 Digital Collections Profile (DCP) and the CIP data model, and discuss issues arising from the analysis. This work assumes the reader has copies of the CIP specification and the DCP specification available. The DCP specification can be obtained from the following URL. http://lcweb.loc.gov/z3950/agency/profiles/digital.html

It should be noted that DCP does not contain any OMT diagrams and the diagrams depicting DCP are based on the author's understanding and may be incorrect in some aspects.

(Sub-sections 2.3.2.1, 2.3.2.2 and 2.3.2.3 where prepared by Lou Reich, CSC/NASA. The remaining two subsections on DCP were developed by G. Percivall.)

2.3.2.1 Background Information

The "Z39.50 Profile for Access to Digital Collections" is currently under development for the US Library of Congress. The intent of the profile is to provide a very high level structure to enable a user to navigate thematically organized, hierarchically structured collections of descriptions of digital and physical objects. Work on the specification of this profile was begun on in September , 1995 and draft version 7 of the digital collections profile (DCP) was released on May 3, 1996. Due to the focus on high level compatibility the DCP has several stated limitations which are intended to be addressed in companion profiles that deal with more specific domains.

- The DCP treats digital objects as atomic, that is, their content is opaque. Thus the profile addresses searching descriptive information rather than searching digital objects.

- The DCP treats descriptive items (e.g., finding aids, cataloging records, exhibition catalogs) as opaque, though clients may have at their disposal helper applications that are able to process or display them.
- The DCP does not model complex relationships among objects of all classes.
- This profile does not address distributed databases. It does, however, address distributed collections that may be distributed over servers. Thus for the set of databases corresponding to a collection, different databases may reside on different servers, but no individual database is distributed

The Catalogue Interoperability Protocol (CIP) is currently under development by the Protocol Task Team (PTT) of the Committee on Earth Observation Satellites (CEOS). The goal of this protocol is to enable uniform access to the earth observation data stored in a variety of data systems developed by diverse international agencies. Work on this protocol was begun in 1995 and version 1.2 of the CIP was released in March 1996. Development of a new version of the CIP, CIP-B, has begun with an expected release date of February, 1997. The CIP is based on Z39.50-1995 information search and retrieval protocol.

The DCP and CIP views of the holdings of an archive are very similar. Both are based on the basic concepts of description information being logically separated from the described digital object, the description records or item descriptors being organized into collections, and the members of a collection being objects or other subcollections. It would be ideal if the CIP could be viewed as a companion profile to the DCP which extends the DCP to include EOS specific descriptive items and methods to retrieve full or subset EOS digital objects. Due to the fact that the CIP and the DCP were developed independently the achievement of this goal might involve change some portions of the current CIP or DCP. The following section begins the analysis of this approach by presenting OMT diagrams for the CIP and DCP data models and discussing issues that arise in the mapping of the CIP data model as a specialization of the DCP object model.

2.3.2.2 Data Model Issues and Analysis

Figures 2-6 and 2-7 are OMT diagrams which are derived from Sections 2 and 4 of the DCP. The DCP combines both a logical view of digital collections and a physical view of data collections. Figure 2-6 is derived from the DCP logical view where collections are composed of objects and subcollections and reference other collections of interest (related collections). The link to the physical model which is shown in Figure 2-7 is the concept of Descriptive Records (both for Objects and Collections) and the concept of a Datastore which is comprised of the set of all the portions of databases that comprise a collection.

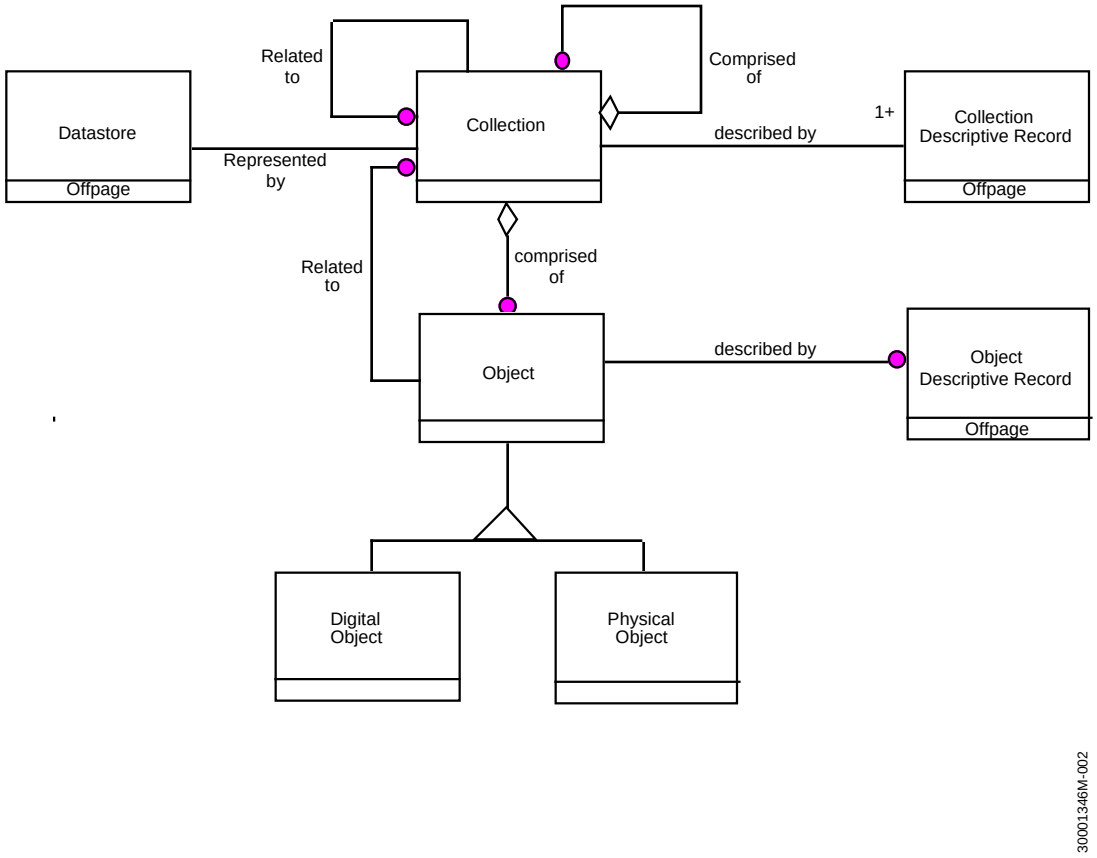

Figure 2-6. DCP: Collection Viewpoint

The central concept of the physical view of the DCP shown is the Descriptive Record. A Descriptive Record exists for a collection and each of the collections and objects are contained within the collection. While a collection is represented as a datastore composed of multiple databases which may on different servers, a Descriptive Record must be contained in a single database and server. A Collection Descriptive Record may enumerate all contained objects and collections. The schema for a Descriptive Record is stated in Section 4 of the DCP. A Descriptive Record contains all the associate descriptions for the object or collection it describes and either a pointer to the object(or collection) or the digital object itself. The Descriptive Record can be considered as a database record, a retrieval record or an abstract database record (schema) depending on the context in which is appears. The DCP states that the distinction between whether something is an object or associated description is dependent on the viewpoint of the collection producer.

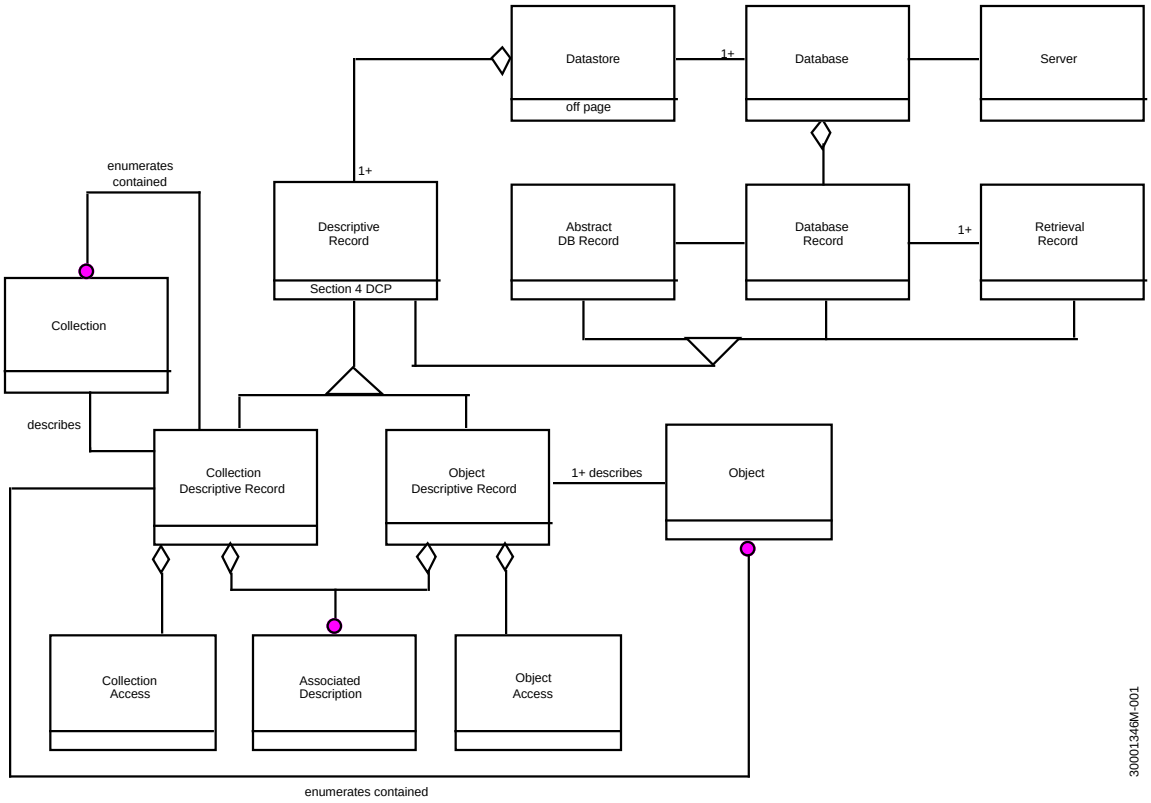

Figure 2-7. DCP: Descriptive Record View

Figure 2-8 is derived from the OMT diagram in Annex F of the CIP specification (If Figure 2-8 is illegible to the reader, please consult the CIP Specification). This model only deals with the model of the CIP objects. These are physical objects such as item descriptors so the CIP view tends to be oriented towards the physical view. It is included for reference.

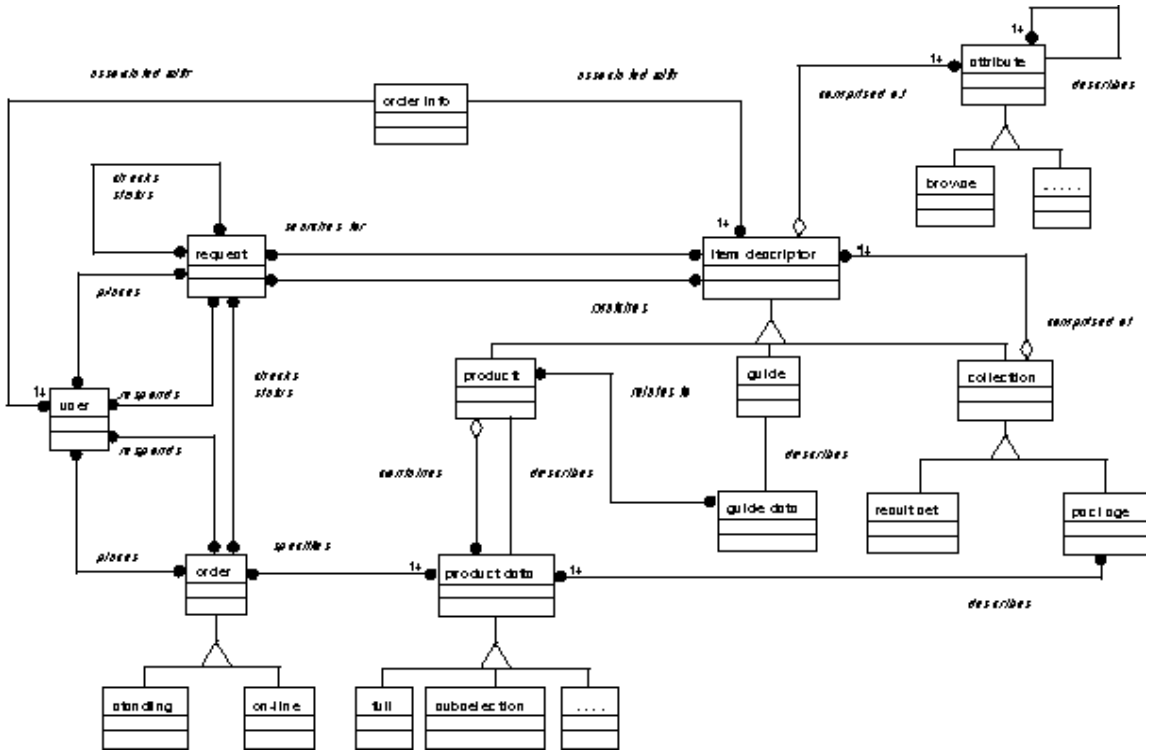

Figure 2-8. CIP- A Object Model

The mapping of the CIP-B data model as a specialization of the DCP object model will be done up when the CIP-B data model is completed. The development of a mapping of CIP-A as a specialization of the DCP involved mapping the CIP item descriptor classes as Associated Descriptions (renamed from Descriptive Items in earlier releases) items in the DCP object model. The CIP product data class was mapped to the DCP digital object class. Some issues that arose during the analysis of CIP and DCP object models include:

1. The CIP logical data model as shown in Figure 2-8 claims to be a logical model of the CIP domain. This does not correlate with the fact that it is based on the physical CIP artifact such as item descriptors. Earlier versions of the DCP object model which combined the logical and physical domain were found to be confusing. CIP-B is developing object models for various viewpoints of the CIP domain. We would recommend that a purely logical viewpoint of the collection domain be included in this effort for the object model. (What is the relationship between DCP datastore and database and the CIP Retrieval manager domain/local site?)

2. The DCP includes related collections so Collection A can refer to Collection B without including Collection B. In the CIP it appears that inclusion is the only possible relationship between collections. The concept of related collections is powerful and should be considered for later releases of CIP.

3. It is not clear how to model guide or browse in the DCP. I would model browse as a Description item while guide might be a related collection or a descriptive item. This issue will need further analysis for clarification.

4. There are some attributes that are duplicated between the CIP and DCP such as collection name and type. Future version of the CIP and DCP would have to be aligned if consistency of attribute sets was a desired characteristic

5. There are many CIP attributes that are valid for collections in any discipline and now would be included in the DCP brief text description. It would be useful if these attributes could be specified in the DCP to facilitate automated searches among differing disciplines.

2.3.2.3 Conclusions

If CIP adopts the direction of being a specialization of DCP the basic object model would need to be revisited. However it must be noted that CIP is a search and order protocol and is based on different set of requirement than the DCP which is a navigational protocol. The DCP is intended to satisfy the navigational and information discovery requirements of a generalist searching through the web with a standard Z39.50 client. However release A of CIP is not oriented towards these casual searches since it involves registering with a member agency and having special client and retrieval manager software. If the later releases of CIP wish to satisfy this type of requirement it will be necessary to do a mapping of the CIP and DCP attribute sets and design the architecture that allows a user to go from a DCP high level scan to a detailed search using CIP. The cost of aligning DCP and CIP is not clear. Though the high level CIP constructs can be mapped to the DCP the effect on the detailed attributes and architecture is far from clear.

Another point that must be considered are the maturity and stability of the attribute sets for CIP and DCP. The latest version of DCP significantly changed the attributes and the usage of Z39.50 Version 3 features. Early alignment of DCP and CIP attribute sets would lower the development impacts and costs but the current DCP does not seem to be a sufficiently stable basis on which to perform this analysis.

2.3.2.4 Recommend CIP/DCP Relationship

This section lists three options and a conclusion for the CIP relationship to the DCP.

1) CIP becomes a specialization of DCP, i.e., CIP uses DCP as a base profile.

The advantage to this approach is compatibility at some level with all others using the DCP profile as well as potential reuse of DCP software developed by others. The cost would be changing the CIP Specification attributes to be compliant with DCP. It is difficult to judge this cost but I suspect it would not be a small effort to convert the CIP spec to DCP for Release B.

The DPRS team could better comment on this. The advantages of this approach are dependent upon the size of the DCP implementation community and the stability of the DCP profile. Unfortunately, I suspect that the DCP community will be small and diverse. Also, the DCP profile is rather new and changes can be anticipated as it is implemented. CIP would have to react to these changes which would divert attention from the mainstream CIP work.

2) CIP adopts DCP concepts and attributes as appropriate

The advantage of this approach is that we would review DCP as it progresses and implement good ideas in CIP as we identify them, e.g., the authoritative attribute. There are several additional concepts which CIP should adopt both in the attributes and the collection concepts. These will be spelled out for discussion in the collection TN. The disadvantage is that we miss out on interoperability with the DCP community. This approach allows CIP to use the good ideas of DCP without being tied to DCP changes.

3) Ignore DCP

This option is included for completeness. The advantage of this option is that it allows all CIP efforts to be focused on mainstream CIP. The disadvantage is that we miss out on good ideas.

Conclusion

It is the conclusion of the PTT that option number 2 above is the appropriate course at this time. The cost of changing the CIP specification to use DCP as a base profile and the subsequent maintenance costs of staying in line with DCP outweigh the advantages of interoperability with DCP implementers. On the other extreme, ignoring DCP would be foolish. So, option number 2 will be followed: track DCP and implement attributes and concepts in CIP as appropriate.

2.3.2.5 Changes to ICS based on Digital Collections

Table 2-3 lists recommended changes to various ICS documents based on the foregoing analysis of the Digital Collections Profile

Table 2-3. ICS Changes Based on Digital Collections Profile

Affected Document	Recommended Change
PTT Development Plan	Add monitoring of Digital Collections as a single task in WBS or as a line item in multiple Work Package Descriptions
CIP Specification	Clarify Logical vs. physical distinction between collections and Databases. Consider DCP logical relationship that descriptive Records describe Collections and Objects. Consider DCP physical relationship that a group of Descriptive Records is a Data Store which is stored in a Data Base.
CIP Specification and Collection Manual	Add DCP concept of "Related Collection." Add attribute of RelatedCollections with relationship values: 1 for superior collection, 2 for context collection, 3 for related collection
CIP Specification and Collection Manual	As decided during the April 1996 PTT Teleconference, the PTT will continue to review the DCP and incorporate ideas into CIP as deemed appropriate. The DCP will not use the DCP as a base profile for Release B.

2.3.3 ECS Collection Model

The ECS Data Model is composed of eight Modules[7]: ECS Collection, ECS Data Granule, Spatial, Temporal, Document, Delivered Algorithm Package, Data Originator, and Contact. Each Module is contains a set of objects. The ECS Collection Module is shown in Figure 2-9. The objects listed as "offpage" are defined in one of the other ECS Modules. Of specific interest to the topics of the CTN is the collection types. As shown in Figure 2-9, the ECSCollection object has two specializations: MultipleTypeCollection and SingleTypeCollection. This section describes the collection types and makes comparisons to the CIP collection types.[8]

[7] Reference: 311-CD-002-004

[8] The next two sections describing the ECS Collection Types are excerpts from a memo under development by Graham Bland for ECS.

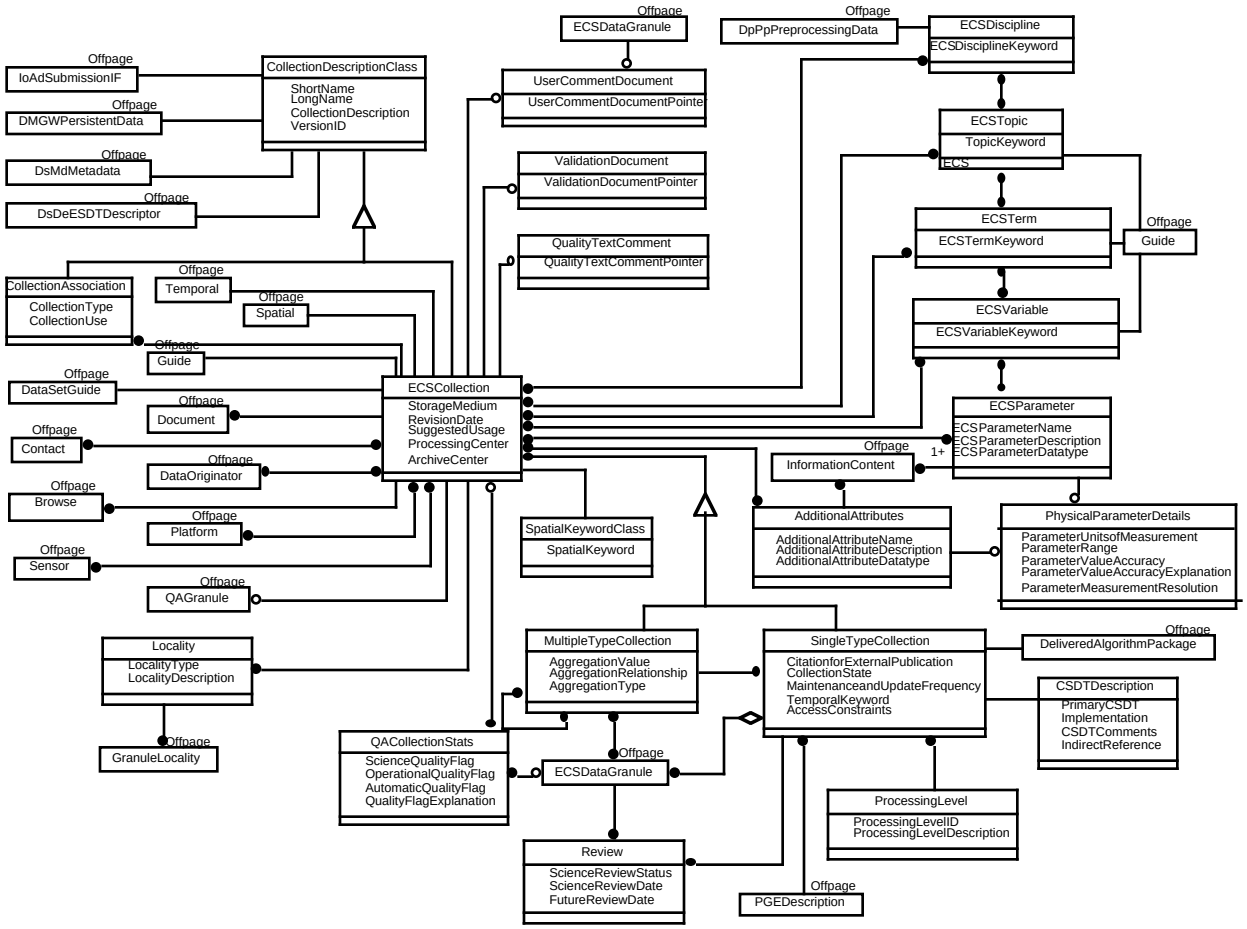

Figure 2-9. ECS Collection Model

2.3.3.1 Single Type Collection

For a collection to be considered single type, the measured quantities must be the same in each product. The measured quantities are the engineering, physical or geophysical measurements contained in each product. Sameness is judged to be related to the characteristics of the measurement; especially the range, units, resolution, reporting precision and known accuracy. These characteristics must be either identical in each product or sufficiently similar as judged by scientific experts to justify the grouping of the products into a single type collection.

A single type collection which is replicated in whole or part, subsetted, subdivided or extended (geographically, temporally) while retaining the same measured quantities may form another version of the same single type collection, or in certain circumstances remain the same version but may not be named a new single type collection. Products from a single type collection may be used in multitype collections; but the products may not be considered as part of other single type collections.

2.3.3.2 MultiType Collection

A MultiType collection covers any type of collection other than a single type including

(a) a collection of products from any source which do not produce the same measured quantities in each product.

(b) a collection of products from several single or multitype collections.

(c) a collection of products and whole collections (single or multitype).

(d) a collection of single type collections.

(e) a collection of multitype collections.

A multitype collection retains all of the defining attribute values of a single type collections from which it is derived but adds the following: AggregationValue, AggregationRelationship, AggregationType

2.3.3.3 Comparison of ECS and CIP Collection Types

ECS defines two types of collections: single type and multitype collections. These definitions are similar to the CIP collections (See Section 2.1.2). Single type collections are very similar to provider archive collections. MultiType collections are similar to provider theme and user theme collections. As both definitions are under development, it will be useful to discuss if the CIP collection definitions could benefit from utilizing some parts of the ECS definitions and vice versa. One result may be that a Provider Archive collection cannot be a mixed collection.

2.3.4 CEO Enabling Services

The Center for Earth Observation is sponsoring several activities which could be used to build later releases of the CIP and ICS. Currently the CEO is sponsoring several proof of concept developments including a CIP-A demonstrator project and a Data and Information Modeling Task. . The CEO has also released an Invitation to Tender for the Enabling Services. The Enabling Services will build on the proof of concept studies.

The CEO Enabling Services contains a Search services/CIP-B component. It is likely that results from the CEO CIP-A Demonstrator (which is a pre-cursor to the Enabling Services) and the Search component itself, will provide very useful results for consideration within CIP Collections. The Enabling Services (of Retrieval Manager, Collections and Explain components) are likely to result in implications for the CIP Collections Model. (At the very least to verify or otherwise its structure and approach).

The CEO Enabling Services development includes a task titled Data and Information Modeling which is required to consider the CIP Release A specification[9]. The total list of items required by the ITT to be considered are:

---

[9] Reference: CEO-ITT

- CEO Information Requirements Document
- Methods and procedures for information modeling
- Preliminary assessment of CIP Release A attributes (from parallel CEO study)
- Technical Annex for the ITT for the design and implementation of the CEO Enabling Services (available June 1996)
- Existent data and information models (e.g. PVL, DIF, AliWeb, etc..)

The model shall reflect the static (*e.g*. formats, description, *etc*.) as well as the dynamic (*e.g.* sources and sinks, information distribution at various servers) view of the European Wide Service Exchange(and its *Successor*) and of the CEO Enabling Services.

The data and information model shall also give an understanding and description of the kind of information that the CEO is going to handle. The model shall reflect the distributed nature of the CEO information servers and shall encompass data replication, information security/identification and information synchronization schemes. A handling of a user profile is therefore necessary.

It is hard to predict how the outcome of the CEO Data Modeling Task might relate to the CIP Collections Model, but several differences can be identified at this time. First, the additional models, e.g., AliWeb, may have an impact on the attributes but not on the collection model. The note on user profile is an item that is not currently in the CIP data model and should be considered.

2.3.5 FGDC

The Federal Geographic Data Committee (FGDC) "Content Standards for Digital Geospatial Metadata" specifies the information content of metadata for a set of digital geospatial data. The purpose of the standard is to provide a common set of terminology and definitions for documentation related to these metadata.[10]

The FGDC standard was used as a source for the CIP Release A attributes and should be reviewed as part of the Release B CIP Specification development. The FGDC standard is silent on the issues of collections and therefore is not considered further in this Technical Note.

2.3.6 GCMD

The Global Change Master Directory (GCMD) offers a comprehensive source of information about worldwide Earth science data holdings available to the science community. The GCMD can be accessed through the World Wide Web(http://gcmd.gsfc.nasa.gov) offering free-text searching using “forms". The GCMD has been operational since 1989 and its database has grown tremendously since its inception to over 2900 directory entries. The GCMD data descriptions include NASA, NOAA, NCAR, USGS, DOE (CDIAC), EPA, and other Federal agency datasets,

[10] Reference: FGDC

along with entries from universities and research centers. The GCMD also contains descriptions of data held outside the U.S. through the International Directory Network (IDN). In addition to the GSFC node, other coordinating nodes are located in Frascati, Italy at the ESA/ESRIN Earthnet Program Office and at the National Space Development Agency Earth Observation Center (NASDA) in Japan. Each node contains an exact copy of the GCMD database which is updated automatically every 2 weeks through an information exchange agreement.

The central aspect of the GCMD database is the high-level data set descriptions that give the user basic information on the data and the point of contact. Each description is in an ASCII-text format called the Directory Interchange Format (DIF). The DIF Template (as retrieved from the GCMD web page on July 3, 1996) is shown in Table 2-XX. In addition, the GCMD maintains a list of Valid Parameter Keywords and a list of Valid Earth Science Location Keywords.

The GCMD DIF was considered when developing the CIP-A attribute set. Further evaluation of differences between the CIP-A attributes and the GCMD DIF will be conducted. For the CIP Collection schema high correlation is desired with the GCMD. This will lessen the effort of the various agencies to populate the CIP collections. The product descriptors will necessarily have some variation, .i.e., directory vs. inventory metadata. The GCMD is not aimed at inventory searches.

The GCMD staff are considering a change to add hierarchical collections to the DIF. This proposal was discussed at the May 1996 Catalog Sub-Group meeting. A hard copy of an e-mail message was distributed, title Interop Listserver Proposal #6. The proposal is to replace the current DIF Filed “aggregated” (see Table XXX) with an Aggregation Group. The aggregation group would contain: Aggregation Criteria, SuperDIFs, and SubDIFs. The idea of Sub-DIFs is the same as collections containing collections in the CIP-A. The concepts of SuperDIFs is similar to the DCP related collection. The aggregation_Criteria is similar to the Commonality attribute described in this TN.

DIF Template

Entry_ID:
Entry_Title:
Group: Data_Set_Citation
  Originator(s):
  Title:
  Publication_Date:
  Publication_Place:
  Publisher:
  Edition:
  Data_Presentation_Form:
  URL:
End_Group
Group: Investigator
  First_name:
  Middle_name:
  Last_name:
  Phone:
  Phone: FAX
  Email: Network > Address
  Group: Address
    This is a free-text field.
  End_Group
End_Group
Group: Technical_Contact
  First_name:
  Middle_name:
  Last_name:
  Phone:
  Phone: FAX
  Email: Network > Address
  Group: Address
    This is a free-text field.
  End_Group
End_Group
Discipline: Valid Discipline Keyword
Parameters: Topic > Term > Variable
    > Detailed Variable
Keyword:
Sensor_Name: SHORT NAME >
    Long Name
Source_Name: SHORT NAME >
    Long Name
Group: Temporal_Coverage
  Start_Date: yyyy-mm-dd
  Stop_Date: yyyy-mm-dd
End_Group
Data_Set_Progress:
Group: Spatial_Coverage
  Southernmost_Latitude:
  Northernmost_Latitude:
  Westernmost_Longitude:
  Easternmost_Longitude:
  Minimum_Altitude:
  Maximum_Altitude:
  Minimum_Depth:
  Maximum_Depth:
End_Group
Location: Valid Location Keyword
Group: Data_Resolution
  Latitude_Resolution:
  Longitude_Resolution:
  Altitude_Resolution:
  Depth_Resolution:
  Temporal_Resolution:
End_Group
Project:
Aggregated:
Group: Quality
  This is a free-text field.
End_Group
Group: Access_Constraints
  This is a free-text field.
End_Group
Group: Use_Constraints
  This is a free-text field.
End_Group
Group: Multimedia_Sample
  File:
  URL:
  Format:
  Caption:
  Group: Description
    This is a free-text field.
  End_Group
End_Group
Originating_Center: SHORT NAME
Group: Data_Center
  Data_center_name: SHORT
    NAME > Long Name
  Data_center_URL:
  Dataset_ID:
  Group: Data_Center_Contact
    First_name:
    Middle_name:
    Last_name:
    Phone:
    Phone: FAX
    Email: Network > Address
    Group: Address
      This is a free-text field.
    End_Group
  End_Group
End_Group
Group: Distribution
  Distribution_Media:
  Distribution_Size:
  Distribution_Format:
  Fees:
End_Group
Storage_Medium:
Catalog_LINK:
Group: Reference
  This is a free-text field.
End_Group
Group: Summary
  This a free-text field.
End_Group
Group: DIF_Author
  First_name:
  Middle_name:
  Last_name:
  Phone:
  Phone: FAX
  Email: Network > Address
  Group: Address
    This is a free-text field.
  End_Group
End_Group
IDN_Node:
DIF_Revision_Date: yyyy-mm-dd
Future_Review_Date: yyyy-mm-dd
Science_Review_Date: yyyy-mm-dd

2.4 Additional Collection Types

The current collection types in the CIP definition of collections (see Section 2.1.2) are: provider archive, provider theme, user theme. This section introduces additional collection types relevant to Release B of the CIP and ICS. Also for Release B, the potential of a collection containing a mix of collections, products and guide items is allowed. This is addressed in the section on mixed collections.

2.4.1 Hot Collections

Hot collections were not required for CIP Release A, it is likely that user theme collections will be largely comprised of hot collections. Further definitions for hot collections are required: when would they be formed, persistent of a hot collection, linking hot collections, declaring a hot collection as public, searching a hot collection..

Definition of Hot collection from URD[11]

> **Hot collection** A hot collection is a temporary list of item descriptors that has been generated during the interaction between a user and a Retrieval Manager. This temporary list of item descriptors can be uniquely identified and operated upon, such as searching on a set of search results rather than the full collection(s) again (see Section 2.1.2: CIP Domain Object Model for further details).

Requirement for Hot Collections from URD[12]:

> UR Id : 2.4
> Source : DPRS TN [R1, Section 4.1.5]
> Priority : 2
> Need : C
> Qualifier : RP
>
> **The CIP shall support 'hot collections'. These are lists of collection descriptors or product descriptors resulting from search queries of collections, upon which further queries can be performed.**
>
> *Note : The term 'hot collection' is analogous t o the concept of 'working set' or 'scratch pad list' found within other domains. It is recognized that for the full support of hot collections, specific resource and performance requirements may need to be specified within a separate subsystem requirements specification for the Retrieval Manager.*

2.4.2 Prepackaged Collections

During the Ottawa meeting, the issue of treating a prepackaged piece of media as a collection was raised. Many data centers have a "greatest hits" volume or CDs with a collection of data on a specific topic of popular user interest. A prepackaged collection may be considered a collection type. Another approach would be to consider the prepackaging aspect of the

[11] Reference: URD-A

[12] Reference: URD-A

collection as order option information an not a separate collection type. A prepackaged collection will contain products which may not be part of a provider archive collection. Other issues included: Address prepackaged media as searchable collections separately represented from separate product archive; Prepackaged collections have only "order" as a operation.

2.4.3 Mixed Collections

An analysis of the CIP Specification just prior to the finalization for Release A was conducted to determine the breakage, if any, with the introduction of Mixed Collections. Mixed Collections - collections containing both products and collections - were not allowed in Release A but are required for subsequent releases. A potential use of mixed collections would be to provide greater flexibility in the construction of user theme collections.

As the result of the analysis the **CollectionContents** element was introduced into the collection schema:

CollectionContents =

0 {**IncludedCollections** }1

0 {**IncludedProducts** }1

IncludedCollections = 1 { ItemDescriptorId } n

IncludedProducts = 1 { ItemDescriptorId } n

In the Release B CIP Specification, one other clarification related to mixed collection will be necessary. The CIP Element "Collection Type" (Tag 2,3) requires the following format" <position>, <category>. Where position is either terminal or non-terminal and category is either provider archive, provider theme or user theme. Presently the specification is not definitive as to the position of a mixed collection as a mixed collection would contain both collections and products. Because the intent of terminal is to indicate that the collection contents cannot be further decomposed, a mixed collection should be described as a non-terminal collection.

Discussion on searching of mixed collections is provided in Section 8.2.

2.5 Proposed ICS Collection Model

Section 2 has reviewed the existing CIP collection model as well as several other models of relevance to collections. It is now time to synthesize a data model to represent future ideas for ICS Collections. Figure 2-10 presents an object model of ideas discussed in this CTN concerning collections. It is anticipated that this model will be considered as the URD Object Model is being developed and that the URD Object Model will be come the authoritative references on ICS objects.

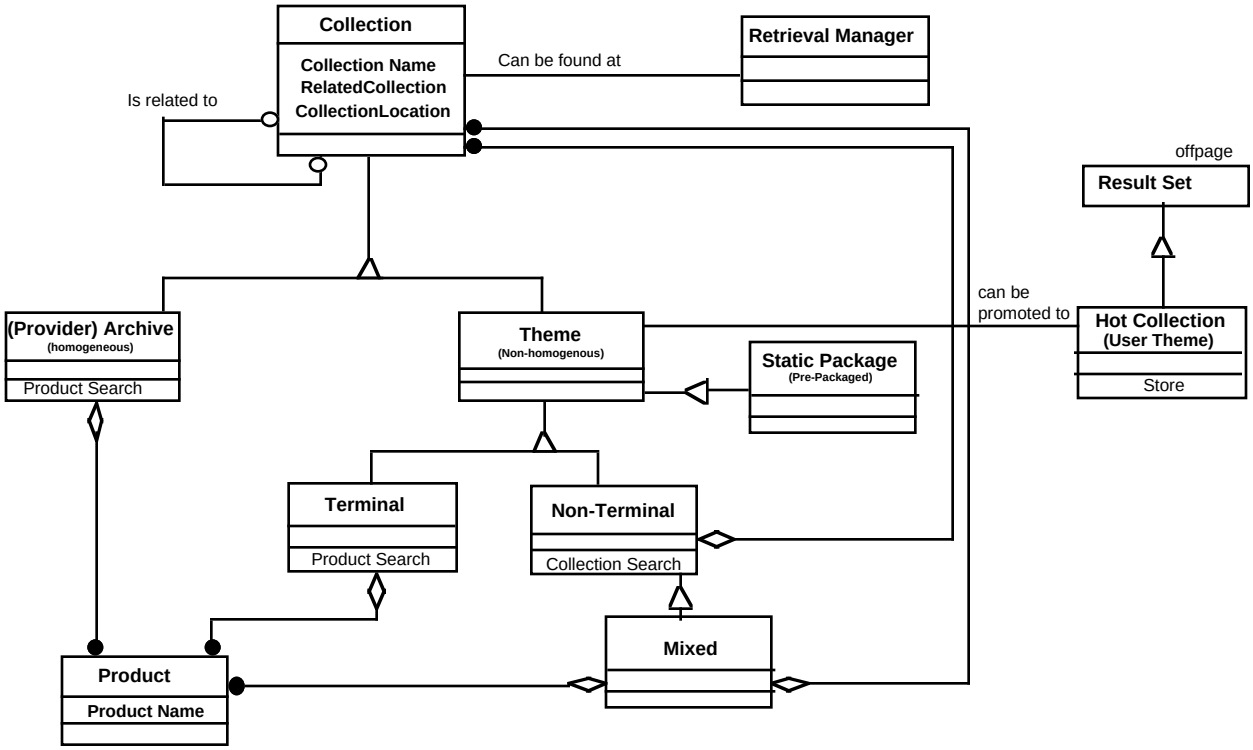

Figure 2-10. ICS Collection Model (Draft)

This page intentionally left blank.

3. Collection Census

3.1 Number of Collections

It order to support the design analysis, a rough order of magnitude estimate of the number of collections which might be accessible by a CEOS CIP system is needed. A specific count of the collections is not necessary nor is a list of named collections. The estimates provided here are also not aimed at indicating an expectation that a particular agency will provide access to the number of collections indicated. The purpose of the estimate is to provide a high end estimate of what the CEOS collection space might grow to. A rough cut at this is developed in Table 3-1.

Table 3-1. ICS Collection Upper Bound for Design Sizing

Agency	Rough Estimate of the Number of Provider Archive Collections	Source of Estimate
European		
BNSC	100 +/- 50	Brian Thomas: Based on a recent survey of UK data providers: 6 Environmental Research Council Data Centers (UKMO, NRSCL, DRA, others). 5 to 10 Retrieval Managers.
CEO	500 to 5000	Ladson Hayes: 500 with small, medium and large providers on board; absolute maximum would be around (5000)
CNES		
DLR		
ESA	50	Christiane Nill: 50 is for ESA provider archive collections only
Canadian		
CCRS	500	Brian McCleod
Japanese		
NASDA	70	Yonsook Enloe
US Agencies		
NASA (EOSDIS)	Version 0: 300 Version 1: 2000	Yonsook Enloe G. Percivall: (V0 collections (via interoperability) +180 V0 migrated data sets[13] + 183 ESDTs) times 3 to account for emergent higher order collections
NOAA	1200	Yonsook Enloe
USGS		
Total	Roughly 10,000	

The estimate reflected in Table 3-1 was discussed during the CEOS Catalog Sub-Group meeting in May 1996. It was the conclusion of the Catalog Sub-Group that the estimate in Table 3-1 significantly under estimated the number of collections which could be index using the CIP. In particular, the GCMD experience indicated that there are millions of datasets, although a large

[13] Reference: Dopplick, Section 3.4 High Priority Data Sets.

portion of these datasets will never be accessible on-line. The number of data sets which may provide on-line directory metadata may be on the order of hundreds of thousands. And for datasets which would have inventory on-line, an estimate on the order of tens of thousands was felt to be accurate.

For now the best that can be said is that CIP should be sized for hundreds of thousands of collections with collection level metadata only with an empty **CollectionContents** attribute. Further CIP should be sized to hold tens of thousands of Collections containing product descriptors.

3.2 Collection Structure Parameters

Also of interest is a projected structure of the total set of collections listed in the previous section. Some key parameters which characterize the structure of the ICS collection space is listed in Table 3-2.

Table 3-2. ICS Collection Structure Parameter SWAGs

Collection Structure Parameter	SWAG
Total Number of Retrieval Managers	Low: 10's based on one Retrieval Manager per agency High: 100's based on many Retrieval Managers contain one or two collections
Avg. Number of collections per collection	Low: 5 High: 50 (May be much higher for a Global collection)
Avg. Collection Depth from Global Collection (see equation below)	Low: 2 to 3 (assumes uniform tree with 30,000 total collections and 50 avg. collections in a collection) High: 6 to 7 (assumes uniform tree with 30,000 total collections and 5 avg. collections in a collection
Average Number of Product Descriptions per Provider Archive Collection	TBD

The following equation was used to calculate several parameters in Table 3-2. For a uniform tree, with D as the collection depth and A as the number of collections per collection, the total number of collection, T, is

$$T = \sum_{i=0}^{D} A^i$$ (see Section 2.2.2.2)

Also of interest is the average number of products in a collection which will need to be estimated based on the type of collection under consideration. It is anticipated that an archive provider collection will have many more (orders of magnitude more?) members than a user theme collection or a hot collection.

3.3 User Model

Also necessary at some point will be an estimate of the user interaction with ICS collections, although this may be more relevant for Retrieval Manager sizing than collection topics.

- Number of users from each agency (outgoing)
- Number of requests per user per year
- Maximum number of requests in an hour (must consider world wide time zones)
- Number of requests for an agency (Incoming)

(This information will be considered in a Future revision of the CTN for ICS/CIP Release C.)

3.4 Metadata Sizing

Also important to ICS element sizing is the size of product metadata, i.e. the size of a product descriptor. This number will certainly vary for the various products. For a first order estimate, an estimate from ECS is provided here. ECS is using an estimate of 2K/product. This estimate is being used for both ECS Release A (predominately Version 0 and TRMM data) and for the ECS DBMS prototype for ECS Release B (Release A plus EOS AM-1, Landsat-7, and others).

This page intentionally left blank.

4. Collection Creation and Maintenance

This section address how the information which is distributed across Retrieval Managers forming the ICS Collection structure will be maintained. First, maintenance concepts from the Release A CIP Specification are reviewed, followed by the introduction of several new collection maintenance concepts. To provide a sketch of how collections will evolve, the following scenarios are presented:

- A Provider Archive Collection is established from an existing archive
- Several local Provider Archive Collections are combined into a Provider Theme Collection
- Key Access nodes are established for a Retrieval Manager
- Remote collections are added to an existing Provider Theme Collection
- Collection maintenance steps, e.g., checking for stale remote links.
- A search result is converted into a user theme collection

After the scenarios, several ideas are presented for automating the collection maintenance. The chapter ends with a taxonomy of procedures which will need to be written to support maintenance activities.

4.1 Collection Maintenance from CIP Specification

The following concepts relating to the creation of collections are taken from the CIP Specification - Release A[14], Section 4.9.2.

There are a number of important assumptions on collection hierarchies and on the Retrieval Manager that are required for effective collection searching:

- collections within a collection tree are defined by the same set of collection descriptor attributes;
- not all collection descriptor attributes for a particular collection node need to have values;
- collections underneath a particular collection node (i.e. subordinate collections) logically belong in that part of the hierarchy, whether by virtue of their common attributes or by virtue of their attribute values, (i.e. collection administrators are building logically consistent collection hierarchies);
- a number of collection descriptor attributes are defined so that if they are included in a search term and a collection node fails to match the search criteria for that search term,

[14] Reference: CIP-A

then any subordinate collections would also fail to match the search criteria (labeled consistency attributes)

The main theme running through these assumptions is that in order that a user and a maintainer can make sense of the collections, collections should be defined and linked based on attributes in the CIP. That is collections should not be established based on concepts which can not be viewed using the CIP. This is both obvious and constraining, but it does limit the manner in which collections can be linked.

4.2 Collection Tree Maintenance Concepts

4.2.1 Generalisation and Collection hierarchy

This section replaces a section titled "Inheritance and Collection Hierarchy" which appeared in Version 0.2 of the CTN. In general, the comments on the Version 0.2 section where unanimous: Inheritance is not the way to build the collection hierarchy. Several reviewers commented that an approach of generalizing the contents of lower level data bases into higher level abstractions is a current area of research. The approach of generalizing existing collections to form a Theme collection which includes the existing set of collections needs to be investigated as part of ICS/CIP Release C.

4.2.2 Commonality

This section contains a **sketchy** proposal about how the concept of commonality could be used to achieve faster and more efficient collection searches[15]. This section is meant to outline the DPRS Team's ideas on this subject, and is intended to spur discussion.

The ideas in this document build on those summarized in the CIP Specification[16] Section 4.9.2, where the notion of consistency attributes is introduced and pruning of collection tree searches by use of consistency attributes. To harmonize conceptual terminology, we have used Commonality attributes here. (Note that consistency attributes are not in any case formally defined in the CIP-A spec!).

4.2.2.1 Overview

The concept of Commonality is an important aspect of collections. Whether this commonality is explicitly expressed (and maybe exploited) by CIP or not, it is in any case embedded within the definition of a collection. A collection shall have commonality with its members by virtue of the choice of the attributes used to define the collection (i.e. a parent collection shall have common attributes, and shall have identical or similar attribute values for at least some of those attributes). This Commonality is the reflection, in CIP terms, of the reason why a collection is created in the

[15] This section was prepared by Marco Ferro and Stuart Mills

[16] Reference: CIP-A

first place (i.e. a collection administrator creates collections, and groups members within these collections, because the parent collection and its children have something in common).

4.2.2.2 Commonality and Collections

A new attribute “Commonality” is defined. This attribute contains a list of all attributes which define the commonality of a collection.

In terms of the CIP Profile, this means that a new Schema Element would be added in the collection Schema:

Commonality ::=

0 { Attribute } n

where “Attribute” identifies an attribute from the CIP Attribute Set as stored in the Explain Database.

Within the definition of a collection, the attribute “Commonality” contains all the CIP attributes (i.e. Search attributes) for which the value of the attribute can be **assumed** to be common in all the children of the collection tree rooted by the collection. The attribute “Commonality” reflects the reason why a collection is defined, and thus contains the attributes for which values are assumed to be “inherited” by all children of the collection.

Example:

To illustrate this, consider the collection tree hierarchy represented in Figure 4-1:

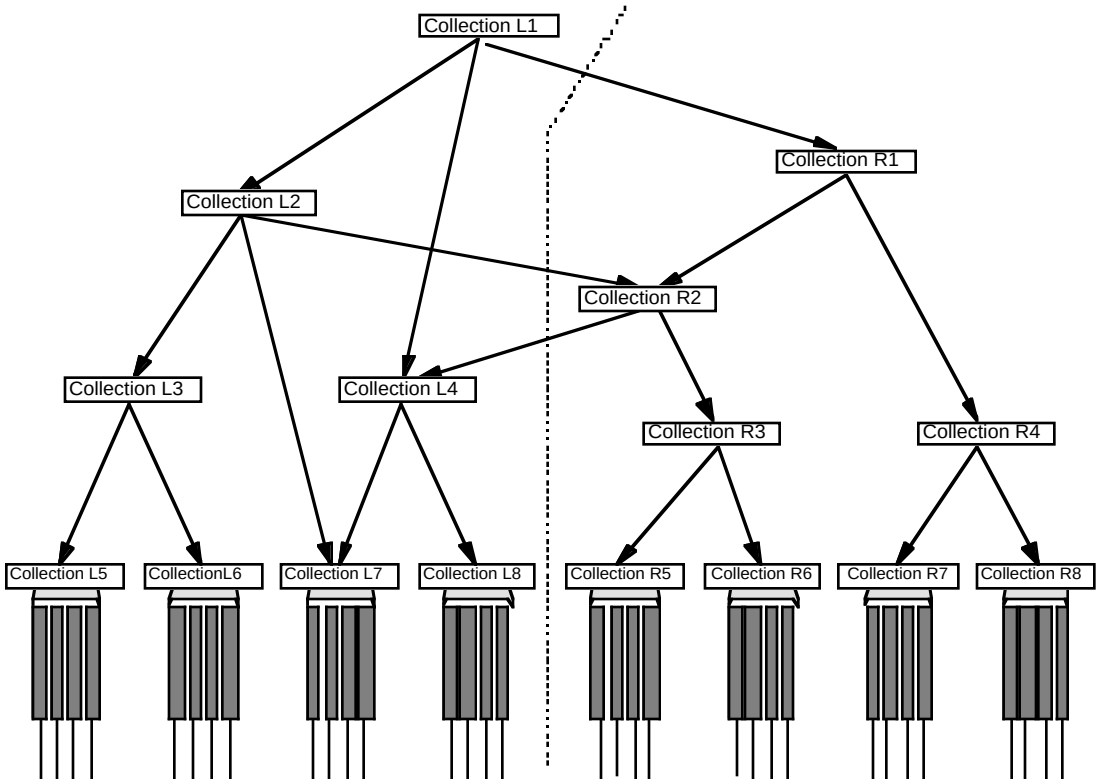

Figure 4-1. Collection Tree

Assume that collection L1 is created for the purpose of containing all SAR images. Its commonality is therefore the value of the attribute “SensorName”, which will therefore be included as value of the “Commonality” attribute.

If a search is targeted at collection L1 by a User, then the User can assume that all the collections below L1 will contain only collections which include the attribute “SensorName” and where the value of “SensorName” will be “SAR” because the attribute “SensorName” is defined as being the commonality of the collection. (Compare this with another collection which contains the “SensorName” attribute with the value “SAR” but where this attribute is not defined as commonality. In this case, the children collections are not assumed to share this attribute, or “inherit” its value).

As consistency would be hard to enforce (because Remote collections may be included somewhere down the collection tree), the CIP has to assume that a collection tree will only contain collections which respect the Commonality declared by the root of the collection tree (e.g. collection R4 may have “SensorName” as Commonality and contain AVHRR images only). This, however, is not a problem. If a collection is created for a specific purpose, then it is a valid to assume that it contains data reflecting this purpose (even if in practice there are anomalies).

4.2.2.3 Commonality and Searches

There is a tradeoff between efficiency and completeness of collection searches: a search tree (i.e. collection tree) can be pruned using the concept of Commonality during searches, however, because of potential inconsistencies within the collection tree this may result in incomplete searches.

For this reason, a client should be able to perform (and choose between) two kinds of searches:

- Common Searches (i.e. searches in which the concept of Commonality is exploited for faster - but potentially incomplete - results)
- Full Searches (i.e. exhaustive searches, including branches that defy commonality)

Because collections are assumed to be consistent, the Commonality information is used by default in searches to prune the search tree. However, if a Full search is desired, this default may be explicitly overridden, in which case all the branches of the search tree will be exhaustively searched.

Note that Commonality is only exploited in a search when the attribute defined as a Commonality attribute in a Collection is included in the search query. If a search query does not contain any attribute defined as Commonality attribute, then obviously no search tree pruning will take place.

Example:

To illustrate this, consider the collection tree hierarchy represented above.

Assume that a user wants to find all SAR images created after January 1 1996. By navigating the collection tree, the User finds that collection L1 contains SAR images of interest. The User would then target its search at collection L1, with a search query.

The following two cases are considered:

- If a Common search is performed, the collection Tree rooted by R4 will not be queried since the Commonality of this tree is “SensorName” and the value for this attribute does not match the value given in the Search Criteria (where SensorName = “SAR”). The search tree can therefore be pruned.
- If a Full search is selected, **all** the collections in the collection tree rooted by L1 will be evaluated. In this case, the search is not as efficient as possible because the collection tree rooted by collection R4 is queried although it contains only AVHRR images.

The distinction between Full and Common search queries could be simply realized with the RPN query language: in the case of complete searches, a flag indicating that the commonality should be overridden could be introduced by appending a simple operand to the original query.

In other words, for the previous example, the Common query would be:

(SensorName = “SAR” AND CreationDate > “01011996”);

and the Full query would be, for instance:

(SensorName = “SAR” AND CreationDate > “01011996” AND-NOT Common).

A possible extension to this concept would allow to distinguish local and wide searches without the explicit use of the EXTERNAL definition. For instance, if all collections at a particular site share as Commonality the host site, than a Full search would correspond to a wide search, and a Common search would correspond to a local search (because all the remote branches of the collection tree (which would be at a different site) would not share the host site Commonality and would therefore be pruned).

4.2.2.4 Open Commonality Issues

1. Which attributes could be used for the definition of Commonality?

Some attributes seem to be good candidates, particularly the Keyword attribute (and its sub-elements) due to the hierarchical nature of its definition (especially if the GCMD valids are enforced), may be more difficult to use (e.g. SpatialCoverage) and should probably be avoided.

2. Commonality Definition

The definition of Commonality for a collection is easy to perform and works well when the Commonality is defined as a conjunction of attributes (e.g. SensorName AND Keyword AND...). However, other kinds of relationship between attribute that it may be desirable to express with Commonality in a collection Tree, such as disjunction (e.g. SensorName OR CreationDate) would be more difficult to include.

3. Commonality and Search Queries:

Search tree pruning is easy to perform with conjunction of search criteria (AND), i.e. if one of the criterion in a search query is not met the query fails and the branch can be pruned. However, conjunctions (and negations) would require more analysis. For instance, if the search query (SensorName = “SAR” OR CreationDate > “01011996”) was evaluated at a collection which Commonality is “SensorName” with the value “AVHRR”, the first part of the query would clearly fail and could be assumed to fail, by Commonality, for subordinate collections. Therefore, the Retrieval Manager could reduce the complexity of the query and forward only a simplified search query (CreationDate > “01011996”) to the subordinate collections. In this way, Commonality is then used to simplify the search query itself and thus increase the efficiency of the searching.. (Note this is likely to be more useful for more complex search queries composed of multiple expressions and predicates).

4. Scope of Commonality

If the ideas about Commonality presented in this document are rejected/not considered by the Collections TN, WE think that the basic idea could still be used and be applied to the GCMD Keywords exclusively.

4.2.3 Guidelines for Defining Key Access Nodes

The Explain database will contain in its **databaseInfo** category the collection tree nodes (i.e. search targets) that are deemed, by the Retrieval Manager Administrator (RMA), as the Key Access Nodes to the collections held in that Retrieval Manager. Whilst there are no mandatory key access nodes, it is anticipated that at least the Retrieval Manager root collection node should be present (this is the node that has no local parent and effectively encompasses all collections owned by a Retrieval Manager).

The following are suggested Key Access Nodes to be established by RMA

- Establish a root Key Access Node for the Retrieval Manager. The root node should (directly or indirectly through other collections) include all collections held by the Retrieval Manager.
- Establish GCMD Roots, i.e., establish higher order collections for all applicable topics in the GCMD master directory[17]: Atmospheric Science, Biosphere, Hydrosphere. Land Surface, Ocean Science, Paleoclimate, Radiance and Imagery, Solar Radiation, Solid Earth, Transient Phenomena
- Establish a key node for data from specific instruments and satellites which is held by the Data Provider, i.e., a collection which contains one collection per satellite held by the data provider.
- Establish a key access node for prepackage collections provided by the Data Provider.

If RMAs uniformly establish Key Access Nodes using these guidelines, Key Access Nodes will be familiar to users across Retrieval Managers. Furthermore, Collections spanning Retrieval Managers will be able to use the uniformly defined Key Access Nodes for establishing higher order collections.

4.2.4 Integration of Existing Schema

One of the key challenges of the RMAs will be to map existing collections with native schema into the CIP Schema. The CIP approach is to have a common data model. The various existing data providers have heterogeneous schemata These data providers will need to map local schema into the CIP Collection and Product schema. The PTT needs to provide at a minimum guidelines for this integration and perhaps even automated approaches for building a schema mapping.

Hughes has investigated how autonomously developed databases can be automatically integrated into a semantic catalog capturing data co-occurrence through a federated database[18]. Using the conceptual graphs knowledge representation paradigm, Hughes has developed an advanced schema integration server that constructs and maintains the integrated semantic catalog.

---

[17] Reference: GCMD

[18] Reference: Dao

Relationships between attributes and attribute domains are identified as the essential semantic information required for completely automated integration to occur. Further investigation will be done to determine the applicability of the Hughes research to the CIP domain.

Plans for the migration of existing data sets into ECS archives should also be investigated for experience with integration of existing schema[19].

4.3 Collection Evolution Scenarios

The scenarios in this section indicate ways in which collections will evolve in the ICS. The initial condition for the following steps is that a data provider has existing archives which are organized (indexed and described) using the schema historically used by the data provider. The Data provider has either built or is reusing Retrieval Manager code and has now hosted that code resulting in an operational Retrieval Manager.

4.3.1 Establishing A Provider Archive Collection

Provider establishes a ICS provider archive collection based on locally existing provider collections, i.e. maps metadata from local collection into CIP attributes. This could be done prior to operations using a static collection or by developing a mapping layer which maps CIP queries in to the local attributes which is passed to the agency archive.

It is anticipated that these initial collections will be single type collections in that the measured quantities will be the same in each product. The measured quantities are the engineering, physical or geophysical measurements contained in each product.

4.3.2 Establishing Higher Level Collections

Using the steps described in the previous section, the provider will have established several Provider Archive Collections based on existing holdings. Now the RMA can establish higher order collections by generalizing on common elements to form Provider Theme collections.

Standard Class Inheritance Templates should be used when establishing higher order collections. Examples of Standard Class Inheritance Templates are shown in Figure 4-2 Class Inheritance Hierarchy Examples".

- Additional topics to be considered: what should be included in a Multi-type collection versus having several subordinate MultiType collections (layering of collections). Issues of how fast to spread collections versus the size of a collection node

4.3.3 Establishing Key Access nodes

Now the RMA will be in a position to establish key Access nodes for the Retrieval Manger. The Guidelines in Section 4.2.3 should be used.

---

[19] Reference: Dopplick

4.3.4 Referencing Remote Collections

To enhance the quality of the Retrieval Manager, the RMA will want to be cognizant of collections in other ICS Retrieval Managers which have related in themes to the collections already present at the local Retrieval Manger.

The RMA can find out Remote Collections on interest either by visiting known retrieval mangers and accessing its Key Access Nodes or by making use of a collection discovery method if it is available (see Section 6).

Remote collections are then added as member to local collections or as related collections using functionality provided by the Retrieval Manager. Guidelines will need to be developed for when to include a collection in the Collection Contents versus when a remote collection should be linked via a related collection attribute.

4.3.5 Collection Maintenance

In order to provide high quality information to the users, the following maintenance steps should be performed on a regular basis:

- Update metadata and schema mapping between internal archives and CIP collections
- Review correctness (consistency, etc.) of Archive Theme Collections.
- Verify all Collection nodes can be accessed from at least the root Key Access Node
- Check for stale links to remote collections.

4.3.6 Converting A Search Result Into a Collection

- To be developed: A search result is converted into a hot collection for an event

4.4 Automated Maintenance Options

This section proposes several options for automating the maintenance of ICS collections. The options come from the Web world of how to deal with stale links. This discussion on maintenance options will result in requirements for collection maintenance and the basis for a trade study on selecting an option.

4.4.1 Maintaining Links using a Spider

One issue with connecting collections in a distributed fashion is that of stale links. Collections may no longer be available when searched by a user. What is needed is an automated means for traversing a web of collections and checking for changes which may require the attention of the human maintainers the collection.

The following article describes a spider which has been developed to automate maintenance of Web Links. As the Web link maintenance problem is similar to the ICS maintenance problem, a

spider like solution may be applicable for ICS. Clearly this easily applies to the maintenance of Related Collections. Further analysis is needed to determine how it applies to Collection Contents. The following is a description of web maintenance and a sketch of a solution.[20]

> Given that a means for automating the traversal process is desired, we need to define the requirements and limitations of such a solution. The primary requirement is that it improve maintenance process by reducing the detrimental effects of human inattentiveness, duplication of effort, and distributed document ownership.
>
> Manual traversal is both time-consuming and boring. Current WWW browsers are designed for the normal viewing process -- they make no distinction between old documents and those that have recently changed, nor do they show the user a document's last-modification and expiration dates. In addition, their only method for testing a link is to actually request and transfer the document contents. This is so inefficient (particularly for sites with slow network connections) that many document owners avoid testing those links at all. Even when applied repetitively (as is required for consistent maintenance), manual traversal fails because no human being can remain consistently attentive during a repetitive, time-consuming, and boring process.
>
> With manual traversal, duplication of effort occurs because different infostructure owners don't see the results of others' traversals. An automated traversal program should therefore be required to handle multiple infostructures, possibly maintained by different owners, and share its testing information across them.
>
> Unfortunately, no automated traversal program can completely solve the maintenance problem. A program cannot tell when a document's contents are changed such that they no longer represent the intentions of a given infostructure. Nor can a program, once it has discovered a broken link, determine why that link is broken or how to fix it. These tasks must still be performed by human maintainers. However, a traversal program can greatly ease the process by alerting the human maintainer and explicitly pointing to those documents that have changed and links that are broken.
>
> Clearly, an automated traversal program would be useful for easing the maintenance of hypertext infostructures. We have developed the Multi-Owner Maintenance spider (MOMspider) for this purpose. MOMspider is a web-wandering robot that, given a list of instructions that details what infostructures to traverse, whom to notify for problems, and where to put the resulting maintenance information, will traverse each infostructure and fulfill all of the requirements listed above.

[20] The material provided is excerpts from an article found at http://www.ics.uci.edu/WebSoft/MOMspider/WWW94/paper.html

4.4.2 Ingrid

The Ingrid approach to constructing links described in the Collection Discover (See Section 6.2.4.1) would benefit the maintenance of collections.

4.4.3 Hyper-G, Hyperwave

Hyper-G is a distributed information system designed to maintain many documents and links. It goes beyond the primitive node-link model used on the Web by offering structuring elements such as collections, clusters and sequences. A description is provided here of Hyper-G[21] based on the commercially available software implementation called HyperWave[22]. Because of Hyper-G's notion of collections it is worthy of further investigation.

Each document has "meta data" associated with it, much like a word processor file's attributes. Fields include title, author, creation time, last modification time, expiration time, keywords, access rights and price. Not only does this simplify electronic commerce and publishing, it permits rapid, accurate searches. It also simplifies server administration, automatically removing expired documents while deleting associated links.

Links are stored in a separate database, simplifying document editing and management and avoiding links to nowhere. With today's Web servers, adding a document to a sequence connected by links typically means editing the surrounding documents and creating all new links. In Hyper-G, you simply insert the document in the sequence, and the links are handled automatically. For the technical details, see "Exploring Hyper-G's Links."

Other advances include improved two-way links between documents and full-text retrieval across an entire server or collection of servers. It supports links to common multimedia file formats and adds support for multiple languages.

One of Hyper-G's greatest strengths is a link update tool called p-flood. This patented algorithm automatically contacts all other Hyper-G servers on the Internet any time a document link from one server to another needs to be updated. P-flood automatically contacts other servers, much like a telephone tree where one server calls a handful of other servers, which, in turn, each call a handful more. In this way, all Hyper-G servers are updated without bogging down the Internet or any server with excess traffic.

4.5 Taxonomy of Collection Maintenance Procedures

This section contains a taxonomy of the collection maintenance procedures which will need to be prepared. The particular contents of the procedures cannot be written until issues in the CTN resolved, but the topic areas in which procedures will be needed can be identified.

The following are the ICS Collection Maintenance Procedures to be developed:

---

[21] Hyper-G, point to http://hyperg.iicm.tu-graz.ac.at/hyperg.

[22] HyperWave Software; http://www.hyperwave.com

- Establishing A Provider Archive Collection
- Mapping a schema on to CIP
- Establishing Higher Level Collections
- Establishing Key Access Nodes
- Referencing Remote Collections
- Review correctness (consistency, etc.) of linked collections
- Verifying Key Access Nodes
- Checking for stale links to remote collections
- Converting A Search Result Into a Collection

5. User Scenarios

This Chapter defines four types of scenarios related to interacting with a Collections Structure:

1) Collection Discovery
2) Collection Navigation
3) Collection Searching
4) Locating Collections with URNs and URLs

These methods are depicted in Figure 5-1 which schematically shows each collection scenario relative to a collection node which is buried in a collection structure. This chapter provides high level scenarios for each method. Subsequent chapters are organized around the four methods and provide requirements, design options and collection model particulars for each method.

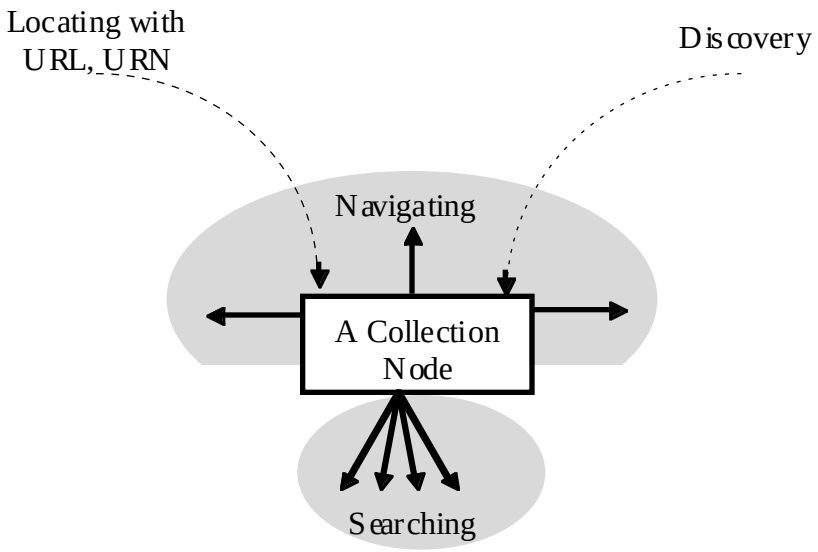

Figure 5-1. Collection Scenario Methods

5.1 Collection Discovery Scenarios

For a user to conduct a CIP search, the user must target a search at collection which is at a particular Retrieval Manager. Before the user can conduct a collection search, the user must find a collection. One approach is to assume that user knows of a Retrieval Manger and through the Key Access Nodes can determine an appropriate Collection at which to target the search. When the assumption that a user will find a collection of interest at the known Retrieval Manager is relaxed, the need to provide a Collection Discovery method arises.

So, the question of this section is: how will a user of ICS, having no prior knowledge of the collection structure, be assured that all collections of interest to the user are examined. A user will need a method to determine all Collections which may be of interest. The discovery of collections will make no assumptions about the user knowing of any existing collections. It is assumed that the user will know of at least one Retrieval Manager.

The user will need to know of at least one Retrieval Manager. Beyond that we should try to lessen the number of assumptions about the users knowledge of the collection structure. For example the user should not have to know about RMs around the world, nor should the user need to know a priori that other RMs hold collections on topics of interest to the user. A method should be provided which allows the user to find all collections of interest across all RMs through a single mechanism - this is termed Collection Discovery. It is this discovery of a collection from outside of the near neighborhood of the collection which is shown in Figure 5-1 as Discovery.

5.2 Collection Navigation Scenarios

Using the terminology of the Digital Collections Profile: The user may select a collection and then navigate to other collections of interest: the client may retrieve a list of related collections, including parent, superior and context collections, brief descriptions of these collections, and descriptions of their relationship to the subject collection. The user might select one of these collections, determine it's parent, superiors and collections.

The collection structure which is navigated through the RelatedCollections element is a directed graph. This is similar to web browsing.

With Navigation, one issue is how a users navigation will result in significant fanning out from the initial collection. Fanout is described by Hinds as assuming a navigator pops up in the collection generalization, then multiple returns from many queries will return[23].

5.3 Collection Searching Scenarios

Collection Searching is the core function of the CIP. A collection is targeted for either a Collection Search or a Product Search. The CIP and Retrieval Managers will then provide searches descending in the collection tree. The following scenarios are variations on collection searching[24].

1) user knows the name of a collection of interest as well as the Retrieval Manager where the Collection Descriptor for the collection resides. The user establishes a session with the retrieval manager and targets a search at the collection.

[23] See discussion in reference: Hinds

[24] Based on scenarios in the Digital Collections Profile.

2) the user may know the name of a collection but not the database where its Collection Descriptor resides. In that case the client may attempt to determine that data base, via Explain[25].

3) It may be that neither the client nor the user knows any collection names, in which case the client might attempt to learn which data bases in general correspond to collections (via Explain) and search those data bases for desired Collection Descriptors. The client may then retrieve Collection Descriptors from these data bases and display summary information to the user, including brief descriptions.

5.4 Collection Names and Location Scenarios

The scenarios in this section assume that the user has a collection name or a collection location in his possession and wishes to find the collection, examine the collection elements, and perhaps target a search at the collection. In the detailed chapter on these scenarios (Chapter 9), URNs will be suggested as mechanism for collection naming. A variant of URLs is already specified in the Release A CIP Specification to specify the location of a Collection.

1) User has a URL for a collection: user can examine the URL and determine information about the collections location, user submits the URL to client and client is able to find collection or find that the collection no longer exists

2) User has the name of a collection: user can examine the name and determine it is a collection name and general topic of the collection. User can submit URN to client which interacts with ICS components and determines zero or more instantiations of the collection.

3) User has a CIP name and can determine that it is a name of a CIP query. User submits query name to a server and the contents of the query are displayed to the user. User submits query and gets results. User sends query to his business partner who later submits query and gets update to results.

[25] See section 5.5 of DCP

This page intentionally left blank.

6. Collection Discovery

The concept of collections is that similar collections are linked by inclusion in other collections. Given that any collection search is always descending in the collection tree, it is not anticipated that an arbitrary collection search will discover all possible collections ICS which could satisfy the query. The result set of a collection query is very dependent upon where the search is initially targeted.

This section address multiple approaches for a CIP user to discover previously unknown collections at previously unknown Retrieval Managers. This is termed Collection Discovery.

6.1 Precedence of Issues

This section describes two issues which are used to categorize collection discovery options which are presented in later sections.

Issue 1. Should the ICS include system level operational elements? A system level operational element is hardware and software that is built and operational specifically for the purpose of the ICS. This system level element would could be used in operational manner by any CEOS agency on behalf of ICS operations. The current architecture in Release A of the CIP Specification does not contain such elements. The current CIP architecture relies on peer-to-peer elements communicating with each other without appeal to a central function. For example there is no element in the Release A architecture which provides a guaranteed view of all retrieval managers. Note that in a non-operational way there is at least one system level element, i.e., the protocol.

CEOS is a federation with each agency bringing its system to play in the CEOS arena. Operationally there should be minimal or no common resource. A true federation has only agency to agency interaction and no element in the CEOS federation required to hold data description of the other agency holdings.

Issue 2. Completeness of Collection Tree. A key feature of ICS is the notion of hierarchical collections or collection trees. This second issue is concerned with the structure of the collection trees in ICS. Depending upon what assumptions are made about the structure of the collection tree determines how a particular collection can be discovered. For example, can we make the assumption that from a tree viewpoint there are no isolated subtrees in the ICS collection structure? Or from a directed graph perspective, can we assume that there are no disjoint clusters? Obviously these assumptions are made, necessary steps would need to be taken in the maintenance of the collections to maintain the assumed structure.

There is linkage between the resolution of these two issues. The order in which these issues are addressed will determine the available choices on issues that are considered later. The precedence that is used in this TN is shown in Figure 6-1.

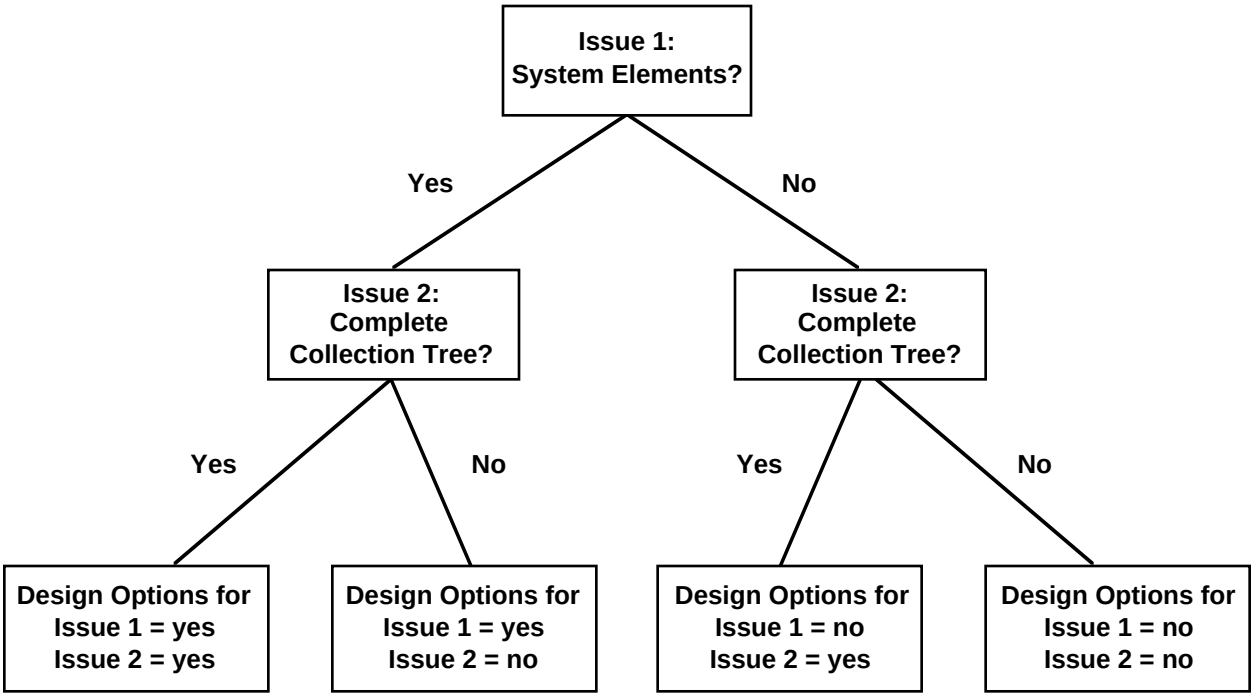

Figure 6-1. Issue Precedence Tree

Clive Best, CEO/JRC, has done a similar analysis as described in this TN and his analysis is reprinted here:[26]

A new supplier of geographic data would need to advertise and reference the resource. There are two ways this could happen.

1) Partly centralized system

   One or more centres (for example the CEO) take responsibility for indexing all the geo-located data. These centres (naming Authorities) assign unique URNs to each resource. The supplier provides the metadata URC at his site for harvesting or sends the metadata to the centre. The integrity of the URC belongs to the supplier who can update the content and resubmit the entry. The advertising of new resources can be automatic or manual. A manual interface could appear as outlined by Vretblad

   All collected metadata would be indexed according to location and time. New entries could be mirrored across the collaborating centres. In addition the text descriptive information would be indexed as is currently done by Lycos, Infoseek and others.

[26] found at http://www.ceo.org/geo_locator.html

Users can search the master index first by location and time as also illustrated by Vretblad. In addition a free text search on the descriptive information would further restrict the search information. A search based on location, time and theme would result in a set of URCs or Metadata. The user can then select the ones of interest and via the URI, connect to the resource itself

The naming authorities themselves would need to be distributed eventually for scaleability reasons (like DNS).

2) Distributed.

The system would adopt a mailing list or the Usenet News model. New resource metadata would be posted to the system. Updates would propagate around the Internet in the same way that Newsgroup updates achieve this. This system would be more complicated than the previous solution, and quality control would be difficult.

6.2 Discovery Options

The options for collection discovery are organized using the issues discussed in the previous section as shown in Table 6-1. The following sections provide a description of each option and, where available, several examples are provided.

Table 6-1. Collection Discovery Options Truth Table

Issue 1:	Issue 2:	
	Collection Tree Complete	Collection Tree Not Complete
ICS System Elements Acceptable	• **Global Collection**	• **Central Index, e.g. Advertising Service**
ICS System Elements Not Acceptable	• **Mobile Agent** • **Distributed Index, e.g. Ingrid, Hyper-G**	• **?**

6.2.1 Global Collection

A Global Collection is a collection which contains **all** ICS collections, either directly or through included collections. The Global Collection would have multiple branches before reaching all terminal collections. To create and maintain a Global Collection would require an effort outside of any one agency to either collect or register new retrieval managers and the contained collections. It may be that the Global Collection is a collection of key access nodes from each of the Retrieval Managers or a collection of all Retrieval Manager root collections. Access to the Retrieval Manager which contained the Global Collection would need to be global to all ICS users.

If a user wished to issue a query which covered all ICS, the Global Collection could be targeted. This Global Search could be a local search which would return the key access nodes for all retrieval managers. Alternatively, the Global Search could be a wide collection search and return all ICS collections which matched the query. Performance of such a global search may be an issue. The schema for the global collection would be identical to the collection schema currently specified (Section 2.1.4)

6.2.2 Central Collection Index

A popular approach to finding Web Resources currently is through one of the many Central Indexes. This notion would be easy to apply to the EO domain. The first two sections describe two Central Indexes for EO data systems currently under development. The later examples of central indexing provided information on how large this design approach could grow based on current web implementations.

6.2.2.1 Advertising Service

An advertising service would not rely on collection searches as the method to find all collections or retrieval managers. An advertising service would allow user searching of a database of data providers or retrieval managers. After a user had used the advertising service to identify data providers of interest, a search could be initiated with a retrieval manager for the data provider site.

Similar to the Global Tree, an Advertising Service would be an element at the ICS system level which would require establishment and maintenance in addition to the agency specific activities required by a ICS site.

An advertising service is part of NASA's EOSDIS Core System (ECS) and is described below. Similar to the ECS Advertising service is CEO's Advertising and Announcement Service. This service will be a successor the current CEO EWSE. EWSE allows users to locate services that are provided by anyone who wishes to advertise. Additional information on ESWE can be found at http://www.ceo.org /ceodocs/ewse.html.

An advertising service is part of NASA's EOSDIS Core System (ECS). The following paragraph is from the ECS Design Specification for the Advertising Service (ECS Document 305-CD-022-001).

> A data provider will advertise its data collections and services with the Advertising Service. The advertisement will include a listing of all products available in the collection and a set of product attributes. Advertisements include directory level metadata, therefore, the attributes reflected in the advertising service include the ECS Core Metadata Directory-Level attributes that apply to collections. The client will send user queries which access only directory level metadata directly to the advertising service (rather than sending it as a distributed query to the various sites which provided the advertising information). A user who wishes to find out what data sets are available on the network can search

(i.e., formulate a query) or browse (i.e., navigate through hyperlinked pages of advertisements) the advertising information. Both types of 'directory searching' are available on the user's desktop; the user can choose whichever approach is most convenient in the current work context.

A early prototype of the ECS Advertising Service can be found at http://epserver.gsfc.nasa.gov:1500

The Advertising Service could be populated in one of multiple ways: each Retrieval manager would publish the links collections of their choosing (similar to the Yahoo central index), versus the advertising service would visit each retrieval manger on a periodic basis[27] and copy collection metadata to advertising service (similar to the Alta Vista and Lycos central indexes).

6.2.2.2 Alta Vista

For estimating the scalability of the advertising service, an existing Web search site such as Alta Vista can be considered. Alta Vista is reportedly the largest central Web Index. The following data was copied from the Alta Vista home page[28]:

ALTA VISTA: THE LARGEST WEB INDEX

Find pointers to your home page; find an old friend; find out what the other indexes missed. You have access to 11 billion words found in 22 million Web pages.

ALTA VISTA: THE FRESHEST NEWS

A full-text index of over 13,000 news groups updated in real-time.

We'll give you the articles too.

ALTA VISTA AND ALPHA: THE FASTEST SEARCH

Alpha 64-bit addressing lets us keep nearly 6 GB of the 33 GB word index in main memory, which makes lookups faster.

6.2.2.3 Lycos

Another current Web central index is Lycos. Lycos is written in Perl, but uses a C program based on CERN's libwww to fetch URLs. It uses a random search, keeps its record of URLs visited in a Perl assoc list stored in DBM. It searches HTTP, FTP, and GOPHER sites, ignoring TELNET, MAILTO, and WAIS. Lycos uses a data reduction scheme to reduce the stored information about each document:

---

[27] It is estimated that to do a full web search using http, visiting every link takes roughly a month - currrently. Reference: conversation with Brad Perry, Hughes Research Laboratory

[28] (http://www.altavista.digital.com/)

* Title
* Headings and Subheadings
* 100 most "weighty" words (using Tf*IDf weights)
* First 20 lines
* Size in bytes
* Number of words

Lycos keeps a word frequency count as it runs...it has read over 25 million words. A list of the most frequent words found after searching 6.3 million words is available off the Lycos home page.[29]

A typical number of query requests per day for Lycos is 200,000[30]

6.2.2.4 WWWW

An existing and perhaps dated approach to Web indexing is the WWW Worm (WWWW) [31]. WWWW is a resource location tool. It is intended to locate all of the WWW-addressable resources on the Internet, and provide a powerful user interface to those resources. The system consists of two parts: one that locates resources, and the other which provides the user interface. A program, wwww, scours the Internet location all web resources - HTML fields and more general URLs. It builds a data base of these. Each HTML file found is indexed with the title attribute used in there.

6.2.2.5 Harvest

A newer central indexing approach is called Harvest[32]. Harvest provides a very efficient means of gathering and distributing indexing information; supports the easy construction of many different types of indexes customized to suit the peculiarities of each information collection; and provides caching and replication support to alleviate bottlenecks.

---

[29] Reference: Lycos

[30] Tomasic, et. al., p5.

[31] Reference: WWWW

[32] This section contains information from the Harvest Home Page (http://harvest.cs.colorado.edu/). Please excuse the sales hype which I have yet to edited out.

Harvest consists of the following subsystems:

Gatherer	The Gatherer provides an efficient and customizable way to collect indexing information using provider site-resident software optimized for indexing.
Broker	The Broker provides an indexed query interface to gathered information. Brokers retrieve information from one or more Gatherers or other Brokers, and incrementally update their indexes. The Broker records unique identifiers and time-to-live's for each indexed object, garbage collects old information, and invokes the Index/Search Subsystem when it receives a update or query. Harvest provides a distinguished Broker instance called the Harvest Server Registry (HSR), which registers information about each Harvest Gatherer, Broker, Cache, and Replicator in the Internet.
Index/Search Subsystem	To accommodate diverse indexing and searching needs, Harvest defines a general Broker-Indexer interface that can accommodate a variety of search engines.
Replicator	Harvest provides a weakly consistent, replicated wide-area file system called mirror-d, on top of which Brokers are replicated. Each mirror-d instance in a replication group occasionally floods complete state information to its immediate neighbors, to detect updates that flood-d failed to deliver, possibly due to a long-lasting network partition, site failure, or failure of a flood-d process. Mirror-d implements eventual consistency: if all new updates ceased, the replicas eventually converge.
Object Cache	To meet ever increasing demand on network links and information servers, Harvest includes a hierarchical Object Cache.
Harvest Object System	Harvest Object System (HOS) allows users to type data and associate method code with the data, which is automatically invoked when a user selects a hypertext link to the data.

Harvest's gathering efficiency derives from a combination of optimized gathering software and a flexible scheme for sharing gathered information among indexes that need it. A Harvest Gatherer collects indexing information, while a Broker provides an incrementally indexed query interface to the gathered information. Gatherers and Brokers communicate using an attribute-value stream protocol called the Summary Object Interchange Format (SOIF), and can be arranged in various ways to achieve flexible and efficient use of the network and servers:

Netscape is proposing the Harvest Summary Object Interchange Format (SOIF) as a standard, as evidenced by their recent decision to adopt SOIF into their product offerings.

To alleviate bottlenecks that arise for accessing popular data and servers, Harvest replicates indexes and caches retrieved objects. The Replication subsystem can also be used to divide the gathering process among many servers (e.g., letting one server index each US regional network), distributing the partial updates among the replicas.

6.2.2.6 GlOSS

Another newer web indexing approach is GlOSS (Glossary-Of-Servers Server). The following is the abstract from a technical paper describing GlOSS[33]:

> As large numbers of text databases have become available on the Internet, it is harder to locate the right sources for given queries. In this paper we present gGlOSS, a generalized Glossary-Of-Servers Server, that keeps statistics on the available databases to estimate which data bases are the potentially most useful for a given query. gGlOSS extends our previous work[34]. We evaluate our new techniques using real-user queries and 53 databases. Finally, we further generalize our approach by showing how to build a hierarchy of gGlOSS brokers. The top level of the hierarchy is so small it could be widely replicated, even at end-user workstations.

6.2.3 Mobile Agents

If a central ICS resource is not assumed but it is assumed that there are no isolated collection trees and disjoint web clusters, then by starting at any Retrieval Manager it is possible to traverse the links between collections and ultimately find any arbitrary collection in ICS. In order to navigate the links mobile agent technology would be applicable.

A mobile agent would move from Retrieval Manager to Retrieval Manager based on the links between collections and searching of collection contents. The agent would need to retain information (or state) about who started the agent and where the agent had previously visited.

Questions of performance would be an issue. The performance could probably be parameterized by the amount of roaming which an agent was asked to do.

Notice that this approach does not rely on any central index resources to be maintained by ICS.

An example of this type of technology is described in the following excerpt from a WWW Conference paper.[35]

> Mobile agents are an emerging technology attracting interest from the fields of distributed systems, information retrieval, electronic commerce and artificial intelligence. We present an infrastructure for mobile agents based on the Hypertext Transfer Protocol (HTTP) which provides for agent mobility across heterogeneous networks as well as communications among agents. Our infrastructure supports the implementation and interoperation of agents written in various languages and takes advantage of current research in HTTP and the World Wide Web in general.

---

[33] Reference: Gravano

[34] Reference: Tomasic

[35] "An HTTP-based Infrastructure for Mobile Agents" found at http://www.w3.org/pub/Conferences/WWW4/Papers/150/.

Recent times have seen exciting new developments in computer networking. Applications like the World Wide Web have made computer networks such as the Internet available (and palatable) to users outside of computer science departments all over the world. Information servers offering all sorts of interesting data are cropping up, and, as researchers are trying to find ways of reliable electronic payment, the net will soon be important as a `virtual marketplace'.

Yet the sheer amount of data available to users in such a network will be difficult to handle. How will they be able to locate the information they need? How are they going to find the best offer for some service they require? One possible solution brought forward to help in this situation consists of `mobile agents' - autonomous programs that move about the network on behalf of their owners while searching for information, negotiating with other agents, or even concluding business deals.

In this paper we propose an infrastructure for such agents. This infrastructure allows agents to move between hosts and communicate with other agents; it supports agents written using diverse languages and lets agent programmers implement a variety of interaction schemes based on a general mechanism for agent communication.

For the purposes of this paper, we assume that an agent is a computer program whose purpose is to help a user perform some task (or set of tasks). To do this, it contains persistent state and can communicate with its owner, other agents and the environment in general. Agents can do routine work for users or assist them with complicated tasks; they can also mediate between incompatible programs and thus generate new, modular and problem-oriented solutions, saving work.

Since agents consist of program code and the associated internal state, we can envision mobile agents which can move between computers in a network. An obvious application of this idea is in information retrieval, where it is easy to picture a mobile agent that gathers interesting data on some computer. If it has gone through all the available data, it moves somewhere else in order to find out even more tidbits before returning to its `owner' loaded with pertinent information. Of course the same information could be retrieved by the owner's computer itself using some suitable mechanism for remote access. The advantage of the agent-based approach is that complex queries can be performed by the agent at the remote side without having to transfer the raw data to the owner's computer first, which would likely waste considerable bandwidth.

6.2.3.1 Telescript

In the last five years General Magic has developed and deployed Telescript, a technology for mobile agents.[36] Active personalized services use General Magic's powerful Telescript

[36] http://www.genmagic.com/index.html

technology--an object-oriented, agent-based language--to turn today's passive networks, like the Web, into active networks. Agents carry the user's personal profiles and perform ongoing autonomous work on the user's behalf--even when the user is off-line.

Telescript agents on the Web are able to find specific information, watch for information, even collect information from multiple Web sites to orchestrate compound tasks, all based on the user's preferences. When the agent completes a user's request, the agent can communicate back to the user in real-time or notify the user off-line via e-mail, pager, fax, or other selected means.

Active personalized services run on standard Web browsers. Telescript engines that are co-resident with Web servers create active Web servers. Mobile agents can move between active Web servers and interact, or simply leverage content and services from conventional passive sites (sites with no Telescript engines). Either way, your active personalized application makes the entire Web appear active to users.

In a centralized application, the user's agents gather information from passive Web sites using HTTP. Agents poll the sites for information, and notify the user when the task is complete.

In a distributed application, the user's mobile agents move ('go') between cooperating active sites (each with its own Telescript engine). The user's agent goes from site to site over TCP/IP connections.

6.2.3.2 Firefly

Firefly is a personal software agent.[37] Firefly is about two simple things: you and the community.

Your agent belongs to you and whenever you use it, it intelligently navigates through the entire firefly community space to discover the information and people who would be of most interest to you. In fact every member of the firefly community has his or her own personal agent, so interacting with firefly is like automating the word-of-mouth process. When you tell your agent what interests you, it goes out and locates those tastes, opinions, preferences and idiosyncrasies most similar. The more you train your agent, the more useful and accurate it gets. The more other people train their agents, the smarter the firefly community.

Firefly currently is an agent to find music and entertainment of interest to people on the Web. It is free to register and get your own agent. I suspect somewhere down the line you must pay.

6.2.4 Distributed Index

This section describes three existing designs for a distributed index. A distributed index does not require a single or replicated central operating element. A distributed index can find any collection in the collection structure even if the collection structure has disjoint clusters.

---

[37] http://www.ffly.com/

6.2.4.1 Ingrid: A Self-Configuring Information Navigation Infrastructure

This section describes Ingrid. Ingrid is a technique for whole-web searching that doesn't require a central search database. Instead, it automatically generates links between similar web resources, resulting in an infrastructure that can be efficiently searched robot-style (at search time). The following material was excerpted from WWW Conference paper describing Ingrid[38]. The Ingrid home page[39] includes an announcement of available software if you participate in their testing.

> This paper presents Ingrid, an architecture for a fully distributed, fully self-configuring information navigation infrastructure that is designed to scale to global proportions. Unlike current designs, Ingrid is not a hierarchy of large index servers. Rather, links are automatically placed between individual resources based on their topic similarity in such a way that clusters of term combinations are formed. The resulting topology can potentially be searched and browsed by a robot efficiently. This paper describes the fundamentals of Ingrid--the topology design and the algorithms for creating and searching the topology. It discusses the scaling characteristics of Ingrid, and gives the scaling results of a limited experiment.
>
> Current browsing on the Web consists of the traversal of 1) hypertext-style links between explicitly related documents, and 2) indexes and meta-indexes, which are usually structured according to organization, sometimes by topic, and are in any event almost always incomplete in their coverage. What is missing is general and complete topic-level browsing--that is, where all resources are linked according to topic.
>
> Current searching on the Web consists of querying single-database search engines. While this method is effective, single-database search engines are necessarily (and usually intentionally) incomplete in their coverage. This is likely to become more rather than less true as the Internet grows. What is missing is complete Internet-wide searching.
>
> The following describes one of many typical usage scenarios for Ingrid. An Ingrid "resource publishing" background process is running in conjunction with a mail archive. When new mail arrives, the Ingrid publisher automatically generates a profile of the mail (author, title, high-weight terms), and sends the profile to the Ingrid forward information server associated with the mail archive. The Ingrid forward information server "inserts" the profile into the Ingrid infrastructure by searching for and attaching links to similar profiles.
>
> Later, a user wishes to find resources related to the topic of the previously inserted mail. Using an Ingrid browser, the user inputs keywords related to the topic. The Ingrid browser launches a robot that, by querying various forward

[38] See http://www.w3.org/pub/Conferences/WWW4/Papers/300/

[39] http://rodem.slab.ntt.jp:8080/home/index-e.html.

information servers, traverses links of the Ingrid infrastructure in search of resource profiles with matching terms. Because of the organization of the links, the robot is able to efficiently find better and better matches. The Ingrid browser presents the best matching resource profiles to the user, along with a set of related terms. The user then expands and focuses his/her search using some of the related terms.

Two things make Ingrid unique and potentially feasible: 1) the logical organization of the Ingrid Topology (and how it leads to efficient navigation), and 2) the algorithm for automatically building the Ingrid Topology. They are described in the following sections.

The Ingrid Topology. The definition of the Ingrid Topology is actually quite simple. Assume a set of Resource Profiles, each with a term combination (set of terms). Each Resource Profile is a node in the Ingrid Topology. Define a cluster as a connected sub-topology. That is, there is a path between any two nodes in a cluster that contain only nodes in that cluster. The Ingrid Topology is a mesh topology whereby for every combination of terms, the Resource Profiles that contain those terms are connected such that they form a cluster.

Algorithm for Automatically Building the Ingrid Topology. This section discusses how to build the "base" Ingrid Topology--that is, how to install the Persistent Forward Information associated with each Resource Profile. In general, the term Ingrid Topology refers only to the Persistent Forward Information.

The basic principle behind installing Persistent Forward Information is simple: When a new Resource Profile needs to be installed, it searches for itself, and then connects to whatever it finds.

The two issues of greatest concern at this point (besides scaling) are 1) how to not be overwhelmed by irrelevant or garbage resources (understanding that one man's garbage is another man's gold), and 2) how to deal with the security problems created by the fact that, in the general case, any FIServer can request any other FIServer to add Persistent Forwarding Information.

6.2.4.2 Hyper-G

Hyper-G also known as HyperWave was described in Section 4.4.3.

6.3 Collection Discovery Metadata

For the central index and distributed index approaches for collection discovery discussed in previous sections, the metadata to be held in the index would need to be defined. One option would be to use the current CIP Collection Schema (see Section 2.1.4) as the definition of the Discovery Index. The next two sections describe two other approaches to collection discovery metadata.

6.3.1 CEO GRC Proposal

For the indexing by some collection discovery methods, metadata for each collection would be needed. The Centre for Earth Observation (CEO) has developed a proposal for the collection metadata format.

The Centre for Earth Observation (CEO) has developed an EO information exchange on the web called European Wide Service Exchange (EWSE). The EWSE has implemented a form of URN/URC type for internal management. The URC has been implemented using the IAFA (Internet Anonymous FTP Archive) format..40 The EWSE format could be extended to include generalized geographic and temporal information. CEO labels this a Geographic Resource Characteristics(GRCs). CEO's suggested format is shown in Table 6-2.

[40] The following is a proposal by Clive Best, CEO/JRC (clive.best@jrc.it) found at http://www.ceo.org/geo_locator.html

Table 6-2. CEO GRC Proposal

```
Template-Type: GEORESOURCE
Name:
URN:
Maintainer:
Address:
Phone:
Contact email:
Access-Method:
Record-First-Verified-Date:
Record-Last-Modified-Date:
Record-Expirary-Date:
Short-Description:
.
URI:
URI:
.
.
###
######  Geographic information - optionally specified by region name
######  or a Lat,Lon Polygon.
######  Height(meters above sea-level) is optional for specialised
######  3-d (e.g. Atmospheric) Applications
###
Geopoint:  [latitude,Longitude,[height]]          (1st point)
Geopoint:  [latitude,Longitude,[height]]          (2nd point Clockwise)
.                              (defines a point, line, triangle or polygon)
Region_name:                   (e.g. global or country, or city>0.3million etc.)
Acquisition-Time:
Start-Time:
End-Time:
Interval:
Data_type:                              (image,scalar,vector)
Physical_parameter:
Parameter_units:
```

The number of Geopoints defines the topology of the spatial area. Extra time parameters are optional. The format should be kept as small and simple as possible. The harvester and search engines could potentially deal with millions of such records.

6.3.2 Centroids as Collections Metadata

For collection discovery, an index is needed to be able to find arbitrary collections of interest. For navigation, knowing which collections are related will allow wondering through the collections like following a path in a directed graph.

Because the hierarchy of collection trees will hide the theme of lower level collections. Centroids provide a way to determine what the hidden metadata is in lower parts of the collection tree. Notice that we cannot use a method like BIND which although it does off load any centralized index server depends unpacking of hierarchical information, i.e., host names. A higher order collection will not contain information about collections contained in the tree but further than

one step lower than the present collection. It would be not be efficient to store all of the lower level collection information in higher level collections. Centroids provide a more efficient approach.

There are several approaches which rely on Centroids[41] one is the Common Indexing Protocol. The Common Indexing Protocol[42] is designed to allow general indexing from most of the attribute-value based directory services. It is an extension of the current WHOIS++ indexing protocol. To participate in an Index Service, that underlying database must also be able to generate a 'centroid', or some other type of forward knowledge, for the data it serves.

The centroid of a server is comprised of a list of the attributes (elements) used by that server, and a word list for each attribute. The word list for a given attribute contains one occurrence of every word which appears at least once in that attribute in some record in that server's data, and nothing else.

For example, if a server contains exactly three records, as follows:

Record 1	Record 2	Record 3
Template: User	Template: User	Template: Domain
First Name: John	First Name: Joe	Domain Name: foo.edu
Last Name: Smith	Last Name: Smith	Contact Name: Mike Foobar
Favorite Drink: Labatts Beer	Favorite Drink: Molson Beer	

The centroid for this server would be

Template: User	Template: Domain
First Name: Joe	Domain Name: foo.edu
John	Contact Name: Mike
Last Name: Smith	Foobar
Favorite Drink: Beer	
Labatts	
Molson	

It is this information which is handed up the tree to provide forward knowledge. As we mention above, this may not turn out to be the ideal solution for forward knowledge, and we suspect that there may be a number of different sets of forward knowledge used in the Index Service. However, the indexing architecture is in a very real sense independent of what types of forward knowledge are handed around, and it is entirely possible to build a unified directory which uses many types of forward knowledge.

---

[41] GlOSS, Hinds

[42] Reference: IETF-comindex

Figure 6-2 illustrates how a mesh of index servers might be created for a set of base servers. Although it looks like a hierarchy, the protocols allow (for example) server A to be indexed by both server D and by server H.

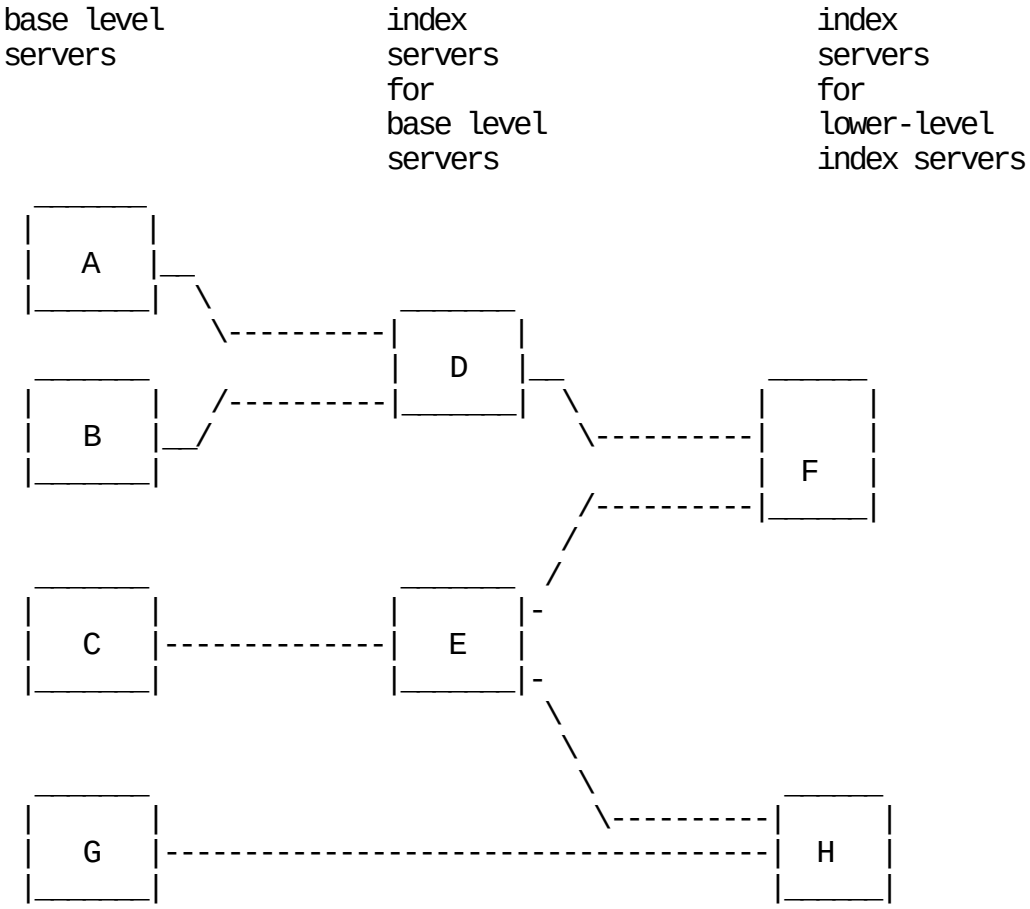

Figure 6-2. Sample layout of the Index Service mesh

In the portion of the index tree shown above, base servers A and B hand their centroids up to index server D, base server C hands its centroid up to index server E, and index servers D and E hand their centroids up to index server F. Servers E and G also hand their centroids up to H.

The Common Index Protocol provides mechanisms for populating the mesh, queries, mesh traversal, loop control, and query referral.

Notice that this approach is similar to CIP in its hierarchical structure, but less data is upwardly propagated in the Centroid approach than say the Global Collection.

6.4 Summary of Discovery Options

This section contains an evaluation of the various options listed above. The assessment is summarized in Table 6-3. The evaluation criteria used here are the recommended URD requirements for Collection Discovery which can be found in an Appendix.

Table 6-3. Summary of Collection Discovery Option Evaluation

Evaluation Criteria	Collection Discovery Options			
	Global Collection	**Central Index**	**Mobile Agent**	**Distributed Index**
Able to discover any collection?	Yes, assuming collection structure is maintained	Yes, assuming collection structure is maintained	Yes, assuming collection structure is maintained	?
Performance ranking: Time to find all relevant collections (1= fastest)	1?	3 Two step process: advertising search, then collection search	4	1?
Scalability:	Difficult due to centralized approached. Achieved with replication	Achievable with current technologies, e.g. Alta Vista	Performance degrades	Seen to be a solution for Web scalability.
Maintenance			No additional maintenance	"Self-Maintaining"
Operations: distributed vs. central	Central, replicated	Central, replicated	Distributed, no central elements	Distributed, no central elements

It appears that existing central indexes approach would easily (down) scale to the sizes in the ICS collection Census. Which means from a scalability and performance point of view, a central index would be preferable. It is not clear that this is the best solution for the federated systems architecture desired for ICS. Clearly the Distributed Index or Mobile Agent approaches are better fit for ICS Federation from this point of view. The distributed index would have a performance advantage over a mobile agent.

It was agreed during the May 1996 PTT meeting that most of the collection discovery mechanisms listed in Table 6-1 are too complicated for CIP Release B. It was agreed that forming a Global Collection via collection maintenance procedures is sufficient for Release B. Too support this approach, collection discovery requirements will be needed in the URD. Concerns were discussed that as the ICS collection structure grows, the performance of a search targeted at a Global Collection may not be acceptable.

6.5 Independent Comments on Discovery

(This section was provided by Nigel Hinds (University of Michigan) as part of his independent critique of Version 0.2 of the CTN.)

The basic discovery problem is that completeness will cost something. It will either cost the user response-time or a sys admin space in which to store indices. Below I've described some trade-offs between completeness, response-time, and space.

Response-time can be reduced by increasing total size of the data through replication and the creation of indices. In a completely distributed environment response-time can be reduced by reducing the number of locations searched (reduced completeness).

Space requirements can be reduced by compression and by eliminating replication. However, this may increase response-time. Space can also be reduced by reducing the total amount of data stored in the space(i.e. reducing completeness).

As EOS and similar information systems become larger these trade-offs will also become critical. Below is a short (and undoubtedly incomplete) list of attributes of an ideal discovery solution.

1. Quantify recall and precision
2. Allow users to specify a recall/precision/response trade-off
3. Maintain discovery structures for each user
4. Allow sharing of discovery structures

Item #1

Relevance weights have been a popular method of identifying and ordering results so the user sees the most useful data first. Assuming that the best answer will do, these methods go a long way to address the problem of completeness. Having said that, relevance weights are generally difficult to produce and evaluate. Also, in the case of a distributed system, the best item (most relevant) may be at an unknown site.

Item #2

Often users are willing to wait for more accurate (complete) results. A good discovery system will allow users to specify an acceptable time/completeness trade-off. The EOS concept of a standing order could be considered an example of such a user trade-off.

Items #3 & #4

Index servers such as Lycos and Yahoo are powerful because they have collected, preprocessed, and centralized data. To scale well, systems should facilitate sharing of indices (meta-metadata). The Harvest model uses this approach. (The Harvest SOIF is analogous to the CIP abstract record structure (Table 2-2)). An even more general approach would be to allow every user to function as a broker if they so desired.

7. Collection Navigation

Collection Navigation is possible if the RelatedCollections Element is added to the CIP collection Schema. (See Section 2.3.2 for a discussion of the related collections from the DCP, and Section 2.5 for a recommendation on incorporating related collections into the CIP object model). Having this information would allow browsing the collection structure in a manner identical to current Web Browsing. This navigation will also be enabled by use of URLs (See Chapter 9).

This page intentionally left blank.

8. Collection Searching

This section describes collection searching using CIP. The first section provides a quick review of the two search types currently specified in the CIP Specification. The second section deals with Mixed Collection searching. As mixed collections will be new, how the protocol is to deal with the searches and results sets need to be defined. The third section deals with Local Attributes describing various approaches to dealing with collections which have attributes different than the CIP elements. Lastly a section on pruning searches based on the inheritance which could be built into the collection trees is discussed.

8.1 Current Searches

8.1.1 Collection Search

This section provides a summary of Collection Searches. See the CIP Specification -Release A[43] for more detail.

By targeting a search at a collection, the Retrieval Manager will automatically search all collections below the target collection, i.e. those that are included hierarchically below the target collection.

Once a collection within a collection tree hierarchy fails to satisfy the search criteria against a consistency attribute, then the Retrieval Manager can stop searching the remainder of the tree underneath this node. Therefore, some 'pruning' of the hierarchy is possible during the collection search.

If a successful match results from searching the attributes of a collection, then the Retrieval Manager will continue to search the attributes of the members of the collection in a similar manner. Note a collection search shall not be executed on collection members which are product descriptors (i.e. terminal collection members). If the collection members are collections owned by another Retrieval Manager, then the search shall be passed onto that Retrieval Manager by the originating Retrieval Manager acting as an *origin* to the new *target* Retrieval Manager. This shall continue until the search is complete.

To identify if a search has already been executed on a collection, each search that is forwarded on to another Retrieval Manager, must include the ***ReferenceId*** established for the original search *operation* in the ***additionalSearchInfo*** field of the search object. This ***ReferenceId*** is unique for any *operation* within and beyond a single *Z-association* so that a remote Retrieval Manager can log which particular searches have 'visited' a collection and if a 'visit' has already

[43] Reference: CIP-A Section 4.9.2 Collection Searches

occurred the remote Retrieval Manager can respond immediately with an appropriate *response* (effectively an 'AlreadySearched' flag). The search *response* will also include an appropriate Z39.50 *diagnostic message*, rather than just a response of 'zero matches' as zero matches could be interpreted as a successful search with no matches.

8.1.2 Product Search

This section provides a summary of Product Searches. See the CIP Specification -Release A[44] for more detail.

For a product descriptor search, the Retrieval Manager shall identify those product descriptors that are within the terminal collections below the target collection and direct the search to the relevant product descriptor database manager (i.e. local catalogue inventory system). If a collection in the collection tree includes a remote collection, then the complete search is passed to the remote Retrieval Manager identified, targeting the remote collection and the same procedure will be followed at the remote Retrieval Manager. The remote Retrieval Manager will return a *result set* to the original Retrieval Manager, which will then compile the results from all sources and make the *result set* available to the client.

8.2 Mixed Collection Searching

How will CIP deal with mixed collection searching? As described in Section 2.4.3, the CIP Collection Schema is able to describe a Mixed Collection. What is not defined is how searches are to act on mixed collections.[45]

As stated in the excerpt form the CIP Specification above: "a collection search shall not be executed on collection members which are product descriptors (i.e. terminal collection members)." With mixed collections this statement is confusing because product descriptors will be allowed to be part of non-terminal collections.

If we assume that a collection search targeted a mixed collection only returns collections the user may miss products of interest. After executing a collection search, the user may then examine the collection tree which was returned and assume that only terminal collections have products. The user then would target product searches at the terminal collections and miss the products in the mixed collections[46].

With the addition of mixed collections, the definition of searching will need to be enhanced. Consider Table 8-1.

---

[44] Reference: CIP-A Section 4.9.3 Product Searches

[45] It was decieded during the May PTT meeting that Mixed Collections should not be included in Release B as the value to the user for the cost of providing them was seen to be minmal.

[46] Question of CIP Specication team: Is there a way to limit a CIP search to the contents of a single, i.e., the target, collection?

Table 8-1. Search Control of Mixed Collections

Collection Category	Mixed Collection?	Collection Search	Product Desc Search	Terminal Collection
Provider Archive	Not allowed	Contents not searched	Content searched	yes
Provider Theme	Allowed	Content searched	Content searched	not necessarily
User Theme	Allowed	Content searched	Content searched	not necessarily

To have a meaningful search of the contents of a mixed collection, it is critical that collections and products have common elements which are also searchable attributes. The following table contains those elements which are common to the schemata for collections and products as defined in Appendix C of the CIP specification. The table also indicates if the elements are mandatory in the schemata. Notice that with the exception of Keywords, the mandatory columns are identical.

Table 8-2. Collection and Product Elements

Element	Mandatory in Collection ?	Mandatory in Product?
Spatial Coverage	yes	yes
Temporal Coverage	yes	yes
Keywords	**yes**	**no**
Archiving Center ID	yes	yes
Processing Center	no	no
Data Originator	no	no

The collection Schema will return ItemDescriptorID. The list of IDs could refer to items of various types. The collection schema allows this but provides no information as to the item type. Item type information would need to be in the ID itself. The present CIP specification does not contain a syntax for Product identifiers.

How a particular client would display a mixed collection is not an issue for the CIP Specification.

8.3 Distributed Searching and Local Attributes

In Version 0.2 of the CTN, this section contained a discussion of using local attributes in distributed searches. From discussions during the May 1996 PTT meeting, it is clear that CIP-A will allow a search using local attributes from a remote Retrieval Manger. Therefore the notions previously described in this section about local attribute propagation are not needed.

8.4 Search Pruning using Inheritance

Comment by DPRS Team: Analyze inheritance of attributes/attributes values in more details, especially in relation to the pruning of a search tree. We think that this issue should be approached considering the use of the GCMD keywords, together with rules and guidelines for the definition of collections. As we thought about this issue in some details during the development of the CIP-A Specification, we think we would have very useful inputs for the author to take into consideration/include in the Collection TN.

See Section 4.2.2.

9. Collection Names and Locations

This section address how an ICS user will locate collections using some piece of data, e.g. a collection URL, that is specific to the collection. The URL may have been obtained through an earlier search or passed to the user by a co-worker.

At the Ottawa PTT meeting, a RID relating to PTT investigation of IETF URN proposals was directed to the Collection TN. This section is the response to the RID. For version 1.0 of the CTN, this section has changed significantly. In Version 0.2 it was anticipated that the IETF and W3C groups would have come to agreement on names and address identifier - URIs, URLs, URNs, URCs - collectively referred to UR*'s. No agreement appears have been reached. The CTN Version 1.0 provides a high level overview of UR* and a recommendation about a specific Z39.50 URL proposal. The portion on URNs has been scaled back as there is nothing to recommend that the PTT make use of until the issues converge.

9.1 IETF Uniform Resource Architecture [47]

Addressing is one of the fundamental technologies in the web. URIs, or Uniform Resource Identifiers, are the technology for addressing documents on the web. It is an extensible technology: there are a number of existing addressing schemes, and more may be incorporated over time.

This is an overview of addressing information, e.g. URLs, URIs, developed by the WWW Consortium. A basic picture is shown in Figure 9-1.

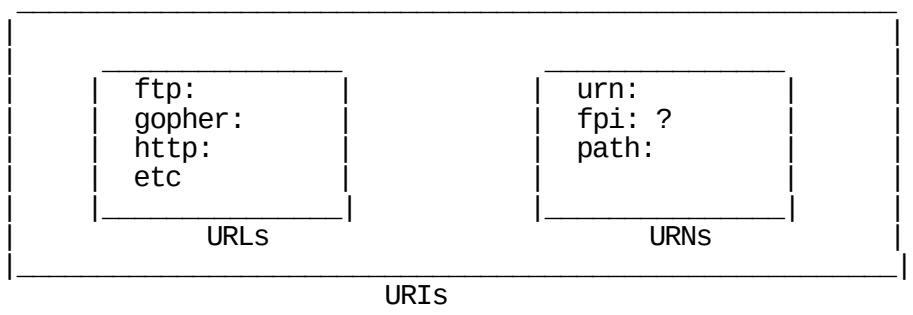

Figure 9-1. Functional Architecture of Uniform Resource Identifiers

Uniform Resource Identifier. The generic set of all names/addresses that are short strings that refer to objects.

[47] This section is based on a web page by the World Wide Web Consortium: url = http://www.w3.org/pub/WWW/Addressing/Addressing.html

Uniform Resource Locators. Exactly what constitutes a locator as opposed to a name is basically lack of persistence, but this is a much discussed point and impossible to define precisely. In practice, the set of schemes referring to existing protocols, listed in the URL specification.

Uniform Resource Name. 1. Any URI which is not a URL. 2. A particular scheme which is currently (1991,2,3,4,5) under development by the IETF, which should provide for the resolution using Internet protocols of names which have a greater persistence than that currently associated with Internet host names or organizations. When defined, a URN(2) will be an example of a URI.

Uniform Resource Citation. A set of attribute/value pairs describing an object. Some of the values may be URIs of various kinds. Others may include, for example, authorship, publisher, datatype, date, copyright status and shoe size. Not normally discussed as a short string, but a set of fields and values with some defined free formatting.

URC is a mechanism of resource description which can be seen as an instance of the general problem of knowledge representation

9.2 URL Type Mechanisms

9.2.1 Current CIP Item Identifiers

Version 1.2 of the CIP specification contains three types of identifiers: DatabaseName, Result Set Name and ItemDescriptorId. Although a format for DatabaseName and Result Set Names are given in the CIP Specification there is a CIP-A RID (CCRS/BM/15) to investigate other mechanisms for naming was assigned to the Collection TN during the Ottawa meeting.

9.2.1.1 Database Names[48]

There are a number of different databases with which a Retrieval Manager must interact. A Retrieval Manager is said to ‘own’ any of the databases that it manages and directly supports. Every Retrieval Manager shall own an Explain database, which will provide the semantics of the information handled by the Retrieval Manager, and the databases corresponding to all the collections within the Retrieval Manager’s domain (i.e. all the collections which are directly managed by the Retrieval Manager)[49].

Whenever the field DatabaseName is required to identify any of the databases owned by the Retrieval Manager it shall be formatted as follows[50] :

Explain database:

---

[48] Reference: CIP-A, Section 4.5.2.3

[49] At present collection databases are the only type of database holding CIP data. It can be envisaged that other types of databases may be added in the future.

[50] This database name follows the standard syntax for URLs.

cip://<rm_ip_address>/IR-Explain-1

CIP database:

cip://<rm_ip_address>/iii_<dddddddd>

where: **iii** = an indicator of the type of database being identified:

iii: **CID** = collection database[51];

rm_ip_address = the fully qualified hostname[52] of the Retrieval Manager that owns the database;

dddddddd = an ASN.1 GeneralString of exactly 8 characters which is unique within the scope of the databases owned by a single Retrieval Manager

9.2.1.2 Result Set Names[53]

Whenever the origin initiates a search, a Result-set-name must be defined so that the result set can be reused later in a Present or Search service. The Result-set-name is defined by the origin and shall follow the format of one of the following two strings:

default indicating the single default result set maintained by the target, or

cip://<origin_ip_address>:<socket>/RSN<dddddddd>&<timestamp>[54]

where :

origin_ip_address = the fully qualified hostname[55] of the machine that originated the search;

socket = the TCP/IP socket number at the origin that the origin has used to connect to the target;

dddddddd = an ASN.1 GeneralString of exactly 8 characters which is unique within the scope of the Z-association.

timestamp = the timestamp of the result set according to the date/time format defined in Ref. CCSDS

Note that all result sets are deleted by the target at the end of the Z-association.

---

[51] At present collection databases are the only type of database holding CIP data. It can be envisaged that other types of databases may be added in the future.

[52] It is recommended that all hostnames be registered as DNS canonical names, instead of absolute IP host name addresses, this avoids problems when Retrieval Managers change their physical machine location and hence absolute IP addresses.

[53] Refernece: CIP-A, Section 4.5.2.4

[54] This result set name follows the standard syntax for URLs.

[55] It is recommended that all hostnames be registered as DNS canonical names, instead of absolute IP host name addresses, this avoids problems when Retrieval Managers change their physical machine location and hence absolute IP addresses.

9.2.2 Uniform Resource Locators for Z39.50

An Internet Draft has been developed which defines URLs for Z39.50 sessions and retrievals[56]. The draft provides the following format for a session and retrieval URL. A Z39.50 Session URL, which opens a client session initialized for interactive use by the user, and a Z39.50 Retrieval URL, which opens and closes a client session to retrieve a specific information item.

As announced by the IETF on 19 June 96 , The IESG has approved the Internet-Draft "Uniform Resource Locators for Z39.50" <draft-ietf-uri-url-irp-05.txt> as a Proposed Standard. This document is the product of the Uniform Resource Identifiers Working Group.

The Z39.50 Session and Retrieval URLs follow the Common Internet Scheme Syntax as defined in RFC 1738, "Uniform Resource Locators (URL)"[57] . In the definition, literals are quoted with "", optional elements are enclosed in [brackets], "|" is used to designate alternatives, and elements may be preceded with <n>* to designate n or more repetitions of the following element; n defaults to 0.

z39.50url = zscheme "://" host [":" port]
["/" [database *["+" database]
["?" docid]]
[";esn=" elementset]
[";rs=" recordsyntax *["+" recordsyntax]]]

Where:

zscheme = "z39.50r" | "z39.50s" (r for retrieval, s for session)

database = uchar

docid = uchar

elementset = uchar

recordsyntax = uchar

and

uchar = alpha | digit | safe | extra | escape (see IETF draft for more details)

Future extensions to these URLs will be of the form of [;keyword=value].

---

[56] Reference: IETF-url-irp-04

[57] Reference" IETF-RFC-1738

9.2.3 Differences between CIP and Z39.50 URLs

Differences between the CIP URLs and the Z39.50 URLs are noted in Table 9-1. For the item for which the definitions overlap (retrieval/result set URL), the structure is significantly different.

Table 9-1. CIP URL versus 39.50 Proposed URLs

CIP, Release A	Z39.50 Proposal	Recommend CIP Changes
Defines syntax of URLs for Collections and Result Sets. (CIP equates collections and databases)	Combined definition for URLs for sessions and retrievals	Add definition for Session URL
Separate formats for collections and result sets	No syntax specified for database name. Single format for session and retrieval URL.	Use single format for session and retrieval URL
Result set is a collection of item IDs based on a previous search request.		
From footnote in CIP Specification: "follows the standard syntax for URLs"	Based on IETF RFC 1738	Verify CIP URLs based on IETF RFC 1738
Result set name is unique across sessions using defined syntax	Uniqueness is not provided	Retain explicit uniqueness
host and port required.	host required, port optional	make port optional

9.3 URN Type Mechanisms

It is interesting to consider the application of the URN/URL/Resolution Service architecture to the naming of Collections in ICS. As presented in the previous section, we currently have a syntax for identifying collection URLs. As is typical of URLs, the collection URL does not meet some of the requirements for URNs which will be discussed in the next section. The Collection URL also leaves too much freedom in the collection name which is embedded in the URL. Ottawa RID /NASA/RM/08 can be answered if a Collection URN syntax can be defined which meets the URN requirements defined by the IETF. Clearly the Resolution Service would be part of the Retrieval Manager functionality, including how a Retrieval Manager could produce a collection URL for any collection URN which was submitted to it.

9.3.1 URN Requirements

The requirements for URNs listed in this section are taken from IETF RFC 1737[58]. RFC 1737 describes a resource, which a URN would be naming, to be either information or objects. The following are Functional Requirements for URNs.

- Global scope: A URN is a name with global scope which does not imply a location. It has the same meaning everywhere.
- Global uniqueness: The same URN will never be assigned to two different resources.
- Persistence: It is intended that the lifetime of a URN be permanent. That is, the URN will be globally unique forever, and may well be used as a reference to a resource well beyond the lifetime of the resource it identifies or of any naming authority involved in the assignment of its name.
- Scalability: URNs can be assigned to any resource that might conceivably be available on the network, for hundreds of years.
- Legacy support: The scheme must permit the support of existing legacy naming systems, insofar as they satisfy the other requirements described here. For example, ISBN numbers, ISO public identifiers, and UPC product codes seem to satisfy the functional requirements, and allow an embedding that satisfies the syntactic requirements described here.
- Extensibility: Any scheme for URNs must permit future extensions to the scheme.
- Independence: It is solely the responsibility of a name issuing authority to determine the conditions under which it will issue a name.
- Resolution: A URN will not impede resolution (translation into a URL, q.v.). To be more specific, for URNs that have corresponding URLs, there must be some feasible mechanism to translate a URN to a URL.

In addition to requirements on the functional elements of the URNs, there are requirements for how they are encoded in a string:

- Single encoding: The encoding for presentation for people in clear text, electronic mail and the like is the same as the encoding in other transmissions.
- Simple comparison: A comparison algorithm for URNs is simple, local, and deterministic. That is, there is a single algorithm for comparing two URNs that does not require contacting any external server, is well specified and simple.
- Human transcribability: For URNs to be easily transcribable by humans without error, they should be short, use a minimum of special characters, and be case insensitive. (There is no strong requirement that it be easy for a human to generate or interpret a URN; explicit human-accessible semantics of the names is not a requirement.) For this reason,

[58] Reference RFC-1737.

URN comparison is insensitive to case, and probably white space and some punctuation marks.

- Transport friendliness: A URN can be transported unmodified in the common Internet protocols, such as TCP, SMTP, FTP, Telnet, etc., as well as printed paper.
- Machine consumption: A URN can be parsed by a computer.
- Text recognition: The encoding of a URN should enhance the ability to find and parse URNs in free text.

9.3.2 IETF Proposals for URNs

Based on the current state of the URN discussions, it is premature to base any PTT conclusions on the IETF proposals.

9.3.3 ECS Universal Reference (URs)

ECS has developed a design framework with similar function as URNs. ECS Universal References (URs) provide ECS applications and users a system wide mechanism for referencing ECS data and service objects. For later CIP/ICS releases, the URs of ECS should be considered.

This page intentionally left blank.

10. Appendix A. Recommended URD Requirements

This appendix contains proposed URD requirements: new requirements and changes to existing requirements based on the topics discussed in the body of the CTN.

The format used for the requirements is that of the URD. An excerpt from Section 3 of URD is provided here:

> Each requirement is uniquely identified (by its 'UR id'), to support forward traceability to subsequent phases.
>
> The requirements are prioritized in terms of the release (B or C, as indicated in the 'Need' entry of each requirement) of the protocol specification to which they are applicable.
>
> There is also a further priority assigned to each requirement, ranging from 1 which is mandatory, through 2 which is important but not essential to 3 which is desirable. This priority is included in the 'Priority' entry of each requirement.
>
> Requirements that require confirmation, e.g. relying on feedback in a subsequent phase (e.g. SR, AD) are marked as 'TBC' and those requiring confirmation of a dependency are marked 'TBD'.
>
> Each requirement has a source stating the origin of the requirement. This may be a reference document, standards, the ESA DPRS project SoW [R2] , name of a recognized body (such as the CEOS or CCSDS), a formal review meeting or 'DPRS team'. This is to support backwards traceability from the URD.

10.1 Data Model for Collections

The RM shall maintain a collection structure

CTN Section 2

Collection structure shall be created and maintained in accordance with ICS Collection Manual

or

There shall be an ICS Collection Manual Created

CTN/PTT Plan

10.2 Collection Creation and Maintenance

This section provides requirements which expand on the current UR Id 69.0 based on the scenarios in Section 4 of the CTN.

10.2.1 Creating Collections

RM shall provide a means for converting a result set into a collection

CTN Section 4

RM shall provide the ability to include local products in a collection

CTN Section 4

RM shall provide the ability to include a local collection in a collection

CTN Section 4

RM shall provide the ability to include a remote collection in a collection

CTN Section 4

RM shall provide the ability to identify a collection which is related to a collection.

CTN Section 4

RM shall provide the ability to identify a collection which is a context collection (i.e. a parent) to a collection

CTN Section 4

RM shall provide the ability to subset a collection and make the subset a new collection

CTN Section 4

10.2.2 Collection Maintenance

RM shall provide the ability to check remote links: manual (B), automated (C)

CTN Section 4

RM shall provide the ability to check consistency of collection structure

CTN Section 4

RM shall provide the ability to check propagation of collection changes across retrieval managers

CTN Section 4

10.2.3 Collection Management

The retrieval manager shall store, maintain, and provide data management services for the collection structure held by the retrieval manager

ECS F&PRS IMS-0220

The RM shall restrict update of the collection structure to authorised users based on user's access privileges.

ECS F&PRS IMS-0230

The RM shall provide at a minimum, data base administration utilities for :

a. Modifying the collection schema

b. Performance Monitoring

c. Performance Tuning

d. Administration of user access control

e. On-line incremental backup

f. On-line recovery

g. export/import of data

ECS F&PRS IMS-0240

The RM shall provide interactive and batch capability for authorised users to add, update, delete, and retrieve information from the RM's collection structure

ECS F&PRS IMS-0260

The RM shall maintain a log of all information update activity

ECS F&PRS IMS-0300

10.3 Collection Discovery

The requirements in this section relate to collection discovery. Collection Discovery is the process by which a user can find collections of interest which would subsequently be targeted for searching using CIP. The requirements in this section are based on the analysis of options in Section 4. These recommended requirements are used to evaluate the options in Section 4. The

requirements of this section are recommended for inclusion in the URD Section 3, although it is not clear how they would be allocated to either the Protocol or Retrieval Manager Sections.

UR Id : 8.1.1
Source : Collection TN [Section 4]
Priority : TBD
Need : B
Qualifier : TBC

The ICS shall provide a service to discover any collection contained in the ICS of the collection using terms which describe the contents of the collection.

As many collections will be unknown a priori to the user, a service by which all collections in the ICS which match the interests of the user will be found. This is a requirement of global discovery of relevant collections in ICS.

UR Id : 8.1.2
Source : Collection TN [Section 4]
Priority : TBD
Need : B
Qualifier : TBC

The ICS collection discovery service shall discover 80% of all relevant collections held in ICS within 10 seconds of submitting a request to the service. This delay shall not include network delay times. The load on the ICS shall be 100 collection discovery service requests per hour during evaluation of this performance.

This requirement is based on the ECS System performance requirement[59] for a directory search using a single keyword attribute. The collection discovery performance is split into two requirements anticipating that users will be satisfied with some results instantaneously and all results eventually. The 80% was chosen based on the 80/20 heuristic.

UR Id : 8.1.3
Source : Collection TN [Section 4]
Priority : TBD
Need : B
Qualifier : TBC

The ICS collection discovery service shall discover all relevant collections held in ICS within 5 minutes of submitting a request to the service. This delay shall not include network delay times. The load on the ICS shall be 100 collection discovery service requests per hour during evaluation of this performance.

This requirement is the second part of the collection discovery service performance. The delay time was chosen by assuming a user would wish to have complete discovery in a typical working session, e.g. 20 minutes. The feasibility of this requirements was determined by assuming multiple iterations of the 80% discovery delay time. For example, tracing a uniform tree of three nodes deep and three nodes per node, requires 12 nodes at 10 seconds each, plus 24 node context switchings at 2 seconds each, resulting in a total of 168 seconds.

UR Id : 8.1.4
Source : Collection TN [Section 4]
Priority : TBD

---

[59] F&PRS, Section 7.5.2.4

Need : B
Qualifier : TBC

The ICS collection discovery service shall function when scaled to the upper bound on ICS Collection census. Collection Discovery performance may be degraded to a doubling of the delay times for this requirement.

10.4 Collection Navigation

The CIP shall support the display and selection of collections related to a current collection which the user has retrieved.

CTN Section 7.

10.5 Collection Searching

Search Pruning using Inheritance, See section 8-4.

10.6 Collection Names and Locations

URN requirements can be divided into requirements on URNs themselves (global scope, location independence, global uniqueness, and persistence), requirements on URN schemes (scalability, legacy support, extensibility, independence of naming authorities, resolvability), and requirements on URN encodings (single encoding, simple comparison, transcribability, transportability, parsability).

This page intentionally left blank.

11. Appendix B. CIP RIDs Mapped to Collection TN

During the Ottawa PTT meeting review of the CIP Specification for Release A, several RIDs on the CIP Specification were to be closed as part of the CTN. This appendix provides a mapping of CIP-A RIDs from the Ottawa PTT meeting as applicable to Collection TN

RID	Topic	Applicable Collection TN Sections
NASA/GP/01	Need to generate separate collection maintenance document	1.2 4
NASA/RM/34	Coordination with z39.50 standards efforts	2.3
NASA/GP/05	Mixed collections	2.4.3 8.2
NASA/RM/06	The collection management will be included in the collection TN.	4
NASA/RM/12	collection naming and collection identification management	9
NASA/GP/08	Key Access Nodes	4.3.3 4.5
NASA/RM/08	collections available form several alternative sites, authoritative attribute	2.3.2
CCRS/DOB/11	It is not clear how the Explain database would work for "hot" collections	2.4.1
CCRS/BM/13	redundancy/caching mechanism	Per PTT Telecon this will not be addressed in CTN
CCRS/BM/14	"mirrored" remote collections	See NASA/RM/08 above
CCRS/BM/15	IETF work on Universal Resource Names (URNs) and Universal Resource Identifiers (URIs)	9

This page intentionally left blank.

12. Appendix C. Release Assignment

This appendix contains the assignment of functionality discussed in this Collection TN to the various ICS/CIP releases. This assignment was developed and reviewed during the May 1996 PTT meeting in Tokyo. The results of these discussions are shown in Table C-1

It was agreed during the Tokyo meeting that for CIP Release B that the mechanism for collection discovery should be through searching a Global Collection (see section 6.2.1). This will allow collection discovery through an existing protocol means, will require collection maintenance procedures to establish the global collection, and may have performance problems when scaled. Other mechanisms may be considered for Release C.

Table C-1 Allocation of Collection TN Topics to Releases

CTN Section	CTN Topic	Release
2.	Data Model for Collections	
2.3.2	Z39.50 Digital Collections Profile	
	Authoritative Attribute	A
	Related collections	B
	Database/Collection Distinction	B
2.3.3	ECS Collection Guidelines for Multi-Type	B
2.4	Additional Collection Types	
2.4.1	Hot Collections: persistent result set	B
2.4.2	Prepackaged Collections	C
2.4.3	Mixed Collections	C
3.	Collection Census	
4.	Collection Creation and Maintenance	
4.2	Collection Tree Maintenance Concepts	
4.2.1	Generalization ~~Inheritance and Collection hierarchy~~	B?
4.2.2	Commonality	B
4.2.3	Guidelines for Defining Key Access Nodes	B
4.2.4	Integration of Existing Schema	C?
4.3	Collection Evolution Scenarios	B - manual C - automated
4.3.1	Promoting a Result Set to a Collection	B
4.4	Automated Maintenance Options	C
5.	User Scenarios: Discovery, Navigation, Searching, Location	
6.	Collection Discovery	
6.2	Discovery Options	B - see below
7.	Collection Navigation	B
8.	Collection Searching	A
8.2	Mixed Collection Searching	C
8.3	Remote Searching and Local Attributes	A
8.4	Search Pruning using Commonality	B
9.	Collection Names and Locations	
9.2	URL Type Mechanisms	B
9.3	URN Type Mechanisms	C

Abbreviations and Acronyms

ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
AVHRR	Advanced Very High Resolution Radiometer
BIND	Berkeley Internet Name Domain
BNSC	British National Space Centre
CCRS	Canada Centre for Remote Sensing
CCSDS	Consultative Committee for Space Data Systems
CEO	Centre for Earth Observation
CEOS	Committee on Earth Observation Satellites
CERN	Conseil Europeen pour la Recherche Nucleaire (European Laboratory for Particle Physics located in Switzerland and France)
CIP	Catalog Interoperability Protocol
CNES	Centre National d'Etudes Spatiales (France)
CSC	Computer Science Corporation
CSMS	Communication and Systems Management Segment (ECS)
CTN	Collection Technical Note
DCP	Digital Collections Profile
DI	Descriptive Items (as used in the DCP)
DID	Data Item Description
DIF	Directory Interchange Format
DLR	Deustche Forschungsanstalt für Luft und Raumfahrt
DNS	Domain Name Service
DNS	Domain Naming Scheme
DPRS	Data Packaging and Retrieval Study
DR	Descriptive Record (as used in the DCP)
ECS	EOSDIS Core System

EO	Earth Observation
EOS	Earth Observation Sciences (a company)
EOSDIS	Earth Observing System Data and Information Systems
ESA	European Space Agency
EWSE	European Wide Service Exchange
F&PRS	Functional and Performance Requirements Specification (for EOSDIS)
FGDC	The Federal Geographic Data Committee,
FTP	File Transfer Protocol
GCMD	Global Change Master Directory
GlOSS	Glossary-Of-Servers Server
GRC	Geographic Resource Characteristics
GSFC	Goddard Space Flight Center (GSFC)
HTML	Hyper-Text Markup language
HTTP	Hypertext Transfer Protocol
IAFA	Internet Anonymous FTP Archive
ICS	Interoperable Catalogues System (CEOS)
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITT	Invitation to Tender
JRC	Joint Research Centre (CEO)
MOMspider	Multi-Owner Maintenance spider
NASA	National Aeronautics and Space Administration (US)
NASDA	National Space Development Agency (Japan)
NOAA	National Oceanic and Atmospheric Administration (US)
OCLC	Online Computer Library Center, Inc
OMT	Object Modeling Technique (Rumbaugh)
PTT	Protocol Task Team
PVL	Parameter Value

RFC	Request For Comment
RID	Review Item Discrepancy
RMA	Retrieval Manager Administrator
SAD	System Architecture Document
SAR	Synthetic Aperture Radar
SDPS	Science Data Processing Segment
SMTP	Simple Mail Transfer Protocol
SOIF	Summary Object Interchange Format (Harvest)
SRF	Server Request Framework
TCP	Transmission control Protocol
URC	Uniform Resource Characteristics
URI	Uniform Resource identifier
URL	Uniform Resource Locators
URN	Uniform Resource Names
URs	Universal References (ECS)
USGS	United States Geological Survey
WGISS	Working Group on Information Systems and Services
WWW	World Wide Web
WWWW	WWW Worm
ZIG	Z39.50 Implementers Group